Minimum I-divergence Methods for Inverse Problems

A Thesis
Presented to
The Academic Faculty

by

Kerkil Choi

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy

School of Electrical and Computer Engineering
Georgia Institute of Technology
December 2005

Minimum I-divergence Methods for Inverse Problems

Approved by:

Dr. Aaron D. Lanterman, Chair
School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. Steven W. McLaughlin
School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. Monson H. Hayes III
School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. Russell M. Mersereau
School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. Ming Yuan
School of Industrial and Systems Engineering
Georgia Institute of Technology

Date Approved: 7 November 2005

This dissertation is dedicated to my wife.

Hyo Jeong Jang:

Thank you for your patience, love, encouragement, and support.

ACKNOWLEDGEMENTS

When I considered the acknowledgements section of my thesis, the person who first came to mind was my wife, Hyo Jeong Jang. I cannot fully express my appreciation to her. I would not have been able to finish this thesis without her dedication. I am so grateful for her trust, encouragement, and most of all, love.

I would like to thank Prof. Aaron D. Lanterman for being a fantastic advisor. Prof. Lanterman always has great suggestions about research. I have always been greatly inspired by his research style, intellectual aspects, and insights. He has also been a good mentor concerning other aspects of life; I have really enjoyed discussing my future with him. I could not have found a better Ph.D. advisor. If I become a professor in the future, I would like to be a creative, kind, and inspiring advisor, exactly like Prof. Lanterman. His creativeness and deep insights made most of this thesis a reality. I also wish to thank him for being so helpful in improving my English, in both writing and speaking. He has often inspired me to be a better writer and speaker.

I also would like to express my gratitude to my thesis committee members, Prof. Monson Hayes, Prof. Steven McLaughlin, Prof. Ming Yuan (who served on the defense committee), Prof. William Hunt (who served on the proposal committee), and Prof. Russell Mersereau (who served on the defense committee), for their guidance. Prof. Hayes' suggestions on studying the effects of noisy measurements in phase retrieval became an important part of this thesis.

I would like to thank my great group-mates, Martin Tobias, Will Leven, Lisa Ehrman, Jason Dixon, Jonathan Morris, and Ryan Palkki. They are not only great friends but also great teachers, especially for my English. I learned many cultural aspects about America from them. I especially appreciate Martin Tobias's persistent help on my English and his general knowledge. I will never forget the good time we had taking many courses together at the beginning of our Ph.D. studies.

I have many other friends to thank for their great support. First, I wish to show my gratitude to Jaemin Shin, who is one of my closest friends. I truly enjoyed every conversation I had with him about research, marriage, the future, English, and all other aspects of life. I am thankful that I could write a journal paper with him; we had a fun time learning things from each other. Second, I thank Jinwoo Kang and Soohyun Bae for helping me refresh my graduate life. I also recall old but good memories about two other unforgettable friends, Raviv Raich and Majid Fozunbal; I am pleased to have co-authored papers with both of them. I have many other friends to thank: Rajbabu Velmurugan, Yeo-Sun Yoon, Jung-Won Kang, Kevin Chan, Paul Hong, Farshid Delgosha, Israfil Bahceci, Volkan Cevher, Ning Chen, Michael Farrell, Vince Emanuele, Jay Jeon, Clyde Lettsome, Rungsun Munkong, Soner Özgür, Maneli Noorkami, Hua Qian, Nazanian Rahnavard, Mina Sartipi, Heejong Yoo, and Nicolas Gastaud, to name a few.

I am grateful to the CSIP staff, Christy Ellis, Kay Gilstrap, and Charlotte Doughty, for relieving me from worrying about administrative issues. I especially thank Christy for her kindness and dedication.

I want to express my appreciation and respect to our CSIP faculty that created and continually improved this unique, nurturing environment at Georgia Tech for all of us.

Finally, I would like to express my deepest gratitude to my family: Yoonwoong Choi and Hyunsoon Lee, my parents, for their endless supports and love; and Jungeun Choi, my sister, for her encouragement to complete the Ph.D.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

Problems of estimating nonnegative functions from nonnegative data induced by nonnegative mappings are ubiquitous in science and engineering. One solution method is to find an estimate that minimizes a discrepancy measure between the collected data and a hypothetical output induced by the associated nonnegative mapping when the estimate is the input. There are several possible choices for discrepancy measures. We choose Csiszár's *I*-divergence, which defines an information-theoretic discrepancy between two nonnegative functions. Csiszár found that minimizing his *I*-divergence is the only choice consistent with certain postulates that may be considered desirable for inference problems subject to nonnegativity (*i.e.*, all the functions involved are nonnegative).

This thesis proposes iterative algorithms for minimizing the *I*-divergence in several inverse problems. Our applications can be summarized along the following three lines:

- **Deautocorrelation:** Deautocorrelation involves recovering a function from its autocorrelation. Deautocorrelation can be interpreted as phase retrieval in that recovering a function from its autocorrelation is equivalent to retrieving Fourier phases from just the corresponding Fourier magnitudes. Schulz and Snyder invented an minimum *I*-divergence algorithm for phase retrieval. We perform a numerical study concerning the convergence of their algorithm to local minima.

  X-ray crystallography is a method for finding the interatomic structure of a crystallized molecule. X-ray crystallography problems can be viewed as deautocorrelation problems from *aliased* autocorrelations, due to the periodicity of the crystal structure. We derive a modified version of the Schulz-Snyder algorithm for application to crystallography. Furthermore, we prove that our tweaked version can theoretically preserve special *symmorphic group* symmetries that some crystals possess. Crystallographers are accustomed to assessing an estimated interatomic structure using the so-called *R-factor*. We numerically investigate whether the R-factor is improved by decreasing

the I-divergence. In particular, we perform this study using real data associated with protein $6PTI$.

Concerning real data, noise always exists and corrupts measurements, and hence estimates as well. We quantify noise impact via several error metrics as the signal-to-ratio changes. Furthermore, we propose penalty methods using Good's roughness and total variation for alleviating roughness in estimates caused by noise.

- **Deautoconvolution:** Deautoconvolution involves finding a function from its autoconvolution. Deautocorrelation and deautoconvolution have inspiringly similar structures in that a function is convolved with itself, except that a reflected version is convolved with itself in deautocorrelation. We derive an iterative algorithm that attempts to recover a function from its autoconvolution via minimizing I-divergence. Various theoretical properties of our deautoconvolution algorithm are derived.

- **Linear inverse problems:** Various linear inverse problems can be described by the Fredholm integral equation of the first kind. The inverse blackbody radiation problem is in this category, with a kernel characterized by Planck's law. This problem is inherently ill-posed because of the kernel characteristics; the naive estimates can be easily destroyed by introducing a slight amount of noise in the measurements. We address this problem by proposing penalized minimum I-divergence methods.

  The output for an input of a Rician communication channel can be described by the Fredholm integral equation of the first kind with a shift-varying kernel. We propose an iterative algorithm for estimating a channel input distribution from the corresponding channel output distribution induced by the underlying integral equation.

CHAPTER I

INTRODUCTION

1.1 Inverse Problems Subject to Nonnegativity

Problems involving the recovery of an original object, blurred by a system, from the blurry data are ubiquitous in engineering and science. In many circumstances, both the function of interest and its blurred version are nonnegative. This dissertation focuses these kinds of problems.

When the blurrinng system h is known and independent of the function of interest f, an input f is related to an output g by the Fredholm equation of the first kind:

$$g(y) = \int_{x \in D(f)} h(x, y) f(x) dx, \qquad y \in D(g), \tag{1}$$

where $D(f)$ and $D(g)$ denote the domains of f and g, respectively. When the system can be expressed by this linear mapping, the problem is often called a positive linear inverse problem [23, 114]. The goal is to reconstruct f from a (possibly noisy) version of g.

When the system h is a function of the input, the data generating mechanism is no longer linear, and hence the problem becomes more complicated and difficult. In this thesis, we specifically focus on the cases where h are shifted versions of the input:

$$g(y) = \int_{x \in D(f)} f(y - x) f(x) dx, \qquad y \in D(g), \tag{2}$$

$$s(y) = \int_{x \in D(u)} u(y + x) u(x) dx, \qquad y \in D(s). \tag{3}$$

Eqs. (2) and (3) are called the autoconvolution of f and the autocorrelation of u, respectively. The problem of recovering f from g is called *deautoconvolution* (see Chapter 6); that of recovering u from s is called *deautocorrelation* (see Chapter 4). The former setting has applications in physics, and the latter in astronomy and chemistry.

1.2 Csiszár's I-divergence

Given two functions (or vectors) a and b, Csiszár's I-divergence is given by

$$I(a||b) = \int_{x \in \mathcal{X}} \left\{ a(x) \ln \frac{a(x)}{b(x)} + b(x) - a(x) \right\} dx, \tag{4}$$

where $\mathcal{X}$ is the set over which a and b are defined. When a and b are both zero, the I-divergence is defined as zero.

The I-divergence is a generalization of the Kullback-Leibler distance. The use of the Kullback-Leibler distance has appeared in various fields such as statistics [38, 62], pattern recognition [53, 54, 58], and spectral analysis [97]. Until Shore and Johnson [52, 98] justified, based on their four consistency axioms, the employment of the Kullback-Leibler distance in reconstruction problems, previous justifications had counted on intuitive arguments involving information-theoretic properties [47]. A limitation of the Kullback-Leibler distance is that it only defines a discrepancy measure between two functions that have the same integral. To compensate for this limitation, Csiszár [23] proposed his I-divergence measure and extended the work of Shore and Johnson to axiomatically justify using his I-divergence in reconstruction problems. Unlike the Kullback-Leibler distance, Csiszár's I-divergence measure can accommodate cases involving two functions that have different integrals. A noticeable result of Csiszár's work is that, if the functions involved are nonnegative, minimizing Csiszár's I-divergence measure is the only choice consistent with a set of intuitive postulates such as regularity, locality, and composition-consistency.

Csiszár similarly found that if the data are real or complex valued, then the typical squared-error criterion is the only choice consistent with his postulates. In this sense, I-divergence may be thought of as playing the role that squared-error usually plays in many classic inverse problem formulations.

1.3 Motivation

Much work has been inspired by Csiszár's results. Snyder *et al.* [107] apply the idea of minimizing Csiszár's I-divergence measure to image deblurring subject to nonnegativity constraints. They proposed an iterative algorithm that gives a sequence of estimates with a nice set of properties such as guaranteed convergence to the global minimum, nonnegativity of every estimate in the sequence, and monotonically decreasing I-divergence. Additionally, they argued that deterministic deblurring problems with nonnegativity constraints can be thought of as statistical estimation problems from "incomplete data" based on an infinite number of observed samples, using the weak law of large numbers.

An important finding in [107] may be summarized as follows. Suppose some data can be modeled as a Poisson point process, and we want to estimate the parameter function of the process under an assumption that the parameter function is an output of a linear system with a known kernel. Assume that infinitely many data samples are available. Then, maximizing the expected value of the loglikelihood of the Poisson data is equivalent to minimizing I-divergence between the measured mean value of the data and the estimated mean of the data, which is an output of a linear system as stated above. A similar idea was investigated in [105].

This notion is studied from a more general perspective and rigorously formalized by Vardi and Lee [114]. Vardi and Lee concluded that the problem of finding a maximum-likelihood estimator from a specific type of incomplete data is equivalent to a particular solution of a linear inverse problem subject to a nonnegativity constraint. The algorithm has been used for deblurring problems in computerized tomography [106].

Expectation-Maximization (EM) algorithms [24] are iterative techniques that attempt to maximize the loglikelihood of incomplete (or indirect [99]) data. In [92,105], the authors found EM algorithms for the Poisson point process data model. Furthermore, they discovered that the asymptotic forms of their EM algorithms, where they assume an infinite number of data points, become iterative algorithms that may minimize Csiszár's I-divergence for the deterministic version of the same data model. While other optimization techniques can also be applied [10], their findings motivate us to focus on developing *EM-like* iterative algorithms (*e.g.*, [84]) that seek to minimize the I-divergence. The advantage of this approach is that we can take advantage of the various developments for EM algorithms [31,41,42,99].

1.4 Thesis Organization

The remainder of this thesis consists of three main parts. Chapters 2-5 discuss deautocorrelation, Chapter 6 is about deautoconvolution, and Chapters 7-8 consider linear inverse problems. Although Chapters 2-5 probably flow best when read in order, the thesis is constructed so that each chapter may be read and understood independently of the others, allowing readers in specifically interested in a specific topic to jump straight to that chapter. Chapters 7 and 8 may be readily interchanged.

One application that can be addressed by minimum I-divergence methods is phase retrieval. Phase retrieval is equivalent to deautocorrelation, which attempts to estimate a function from its autocorrelation. Schulz and Snyder noted this fact and derived an algorithm for phase retrieval that tries to minimize the I-divergence.

In Chapter 2, we derive a tweaked version of the Schulz-Snyder algorithm for application to x-ray crystallography (or other applications with periodic structures). In particular, a symmorphic-group preservation property of our tweaked algorithm is discussed. Its practical roles are illustrated via various numerical experiments.

Chapter 3 discusses some practical concerns that arise when applying the minimum I-divergence algorithm to crystallography.

Unfortunately, both the original Schulz-Snyder algorithm and our tweaked version are not guaranteed to end up with "correct" answers for every possible starting point. Chapter 4 discusses issues concerning the algorithm converging to local minima from a numerical viewpoint.

In practice, measurements are always corrupted by noise. Hence, it is important to investigate what impact noise may have on estimates in phase retrieval. Chapter 5 discusses noise artifacts in phase retrieval using minimum I-divergence methods; we also propose penalty methods to alleviate these artifacts.

We observed that deautocorrelation and deautoconvolution have inspiringly similar underlying structures; hence, Chapter 6 addresses the deautoconvolution problem using the minimization of I-divergence.

The inverse blackbody radiation problem can be described by a Fredholm integral equation of the first kind. Chapter 7 addresses this problem based on minimizing the I-divergence between the measurements and hypothetical measurements induced by the integral equation when the estimates are input to the equation.

In a communication research, we are interested in finding an input distribution to a channel that yields a desired output distribution. The channel mappings are often not shift-invariant. The channel mapping can be viewed as a Fredholm equation of the first kind. We apply a minimum I-divergence method and derive an iterative algorithm to solve this inverse problem in Chapter 8.

Chapter 9 concludes this thesis and discusses some possibilities for future work.

CHAPTER II

SYMMORPHIC-GROUP PRESERVING MINIMUM I-DIVERGENCE METHODS FOR X-RAY CRYSTALLOGRAPHY

2.1 Introduction

Determining molecular structures is a common, but essential, task in many fields ranging from Chemistry to Pharmaceutics. X-ray crystallography is a frequently employed tool.

The diffraction data in x-ray crystallography consists of Fourier magnitudes. These diffraction data can be manipulated to produce another informative function, the so-called Patterson function. The Patterson function contains many useful pieces of information on the molecular structure [111].

A Patterson function can be thought of as an "aliased" autocorrelation, which is the correlation of a function with itself. The function is aliased because crystallized molecular structures, represented as *electron density maps*, form periodic functions. Schulz and Snyder created an algorithm that attempts to recover a function from its autocorrelation, although the autocorrelation is not aliased in their application [93]. Luckily, despite the introduction of aliasing, we can readily modify their algorithm so that it can be applied to x-ray crystallography. This chapter discusses this new application of the Schulz-Snyder algorithm. To the best of our knowledge, we are the first to apply the Schulz-Snyder algorithm to x-ray crystallography.

Note that recovering a function from its autocorrelation is equivalent to retrieving Fourier phases from only Fourier magnitudes, since the autocorrelation is the inverse Fourier transform of the squared Fourier-magnitude diffraction data. Therefore, the Schulz-Snyder algorithm is a phase retrieval algorithm. To solve x-crystallography problems, many researchers have tried to apply existing phase retrieval algorithms such as Fienup's algorithms, which seem to be the most popular in the phase retrieval community. Our approach differs from these the variations of Fienup's algorithms [33] in that the Schulz-Snyder algorithm performs all its operations only in the spatial domain, in contrast with the alternating spatial-Fourier projections employed by Fienup's algorithms.

A useful piece of information that can be easily extracted from diffraction data is the *space group* of the electron density map. Space groups are special types of symmetries. We prove that our modification of the Schulz-Snyder algorithm for x-ray crystallography theoretically preserves space groups under some simple conditions. Furthermore, we discuss how this property can be used in practice to naturally incorporate extracted space group information in our estimates.

This chapter is structured as follows. Section 2.2 discusses some background on relevant subjects. In Section 2.3, a simple modification of the Schulz-Snyder phase retrieval algorithm is derived to apply to periodic functions, as in x-ray crystallography. Section 2.4 discusses the symmorphic-group preservation property of our modified algorithm; we also discuss how to use this property to incorporate known symmorphic-group information. Proofs are given in the appendices. Experimental results are presented and analyzed in Section 2.5. Our discussion concludes in Section 2.6.

2.2 Background

2.2.1 X-ray Crystallography

There is currently no device that can directly see the interatomic structure of a molecule. X-ray crystallography is one technique used to find the interatomic structures of molecules. X-ray diffraction data give the magnitudes of the Fourier transform of the crystal structure. We can experimentally measure the magnitudes, but not the phases. The main challenge of x-ray crystallography is to infer the Fourier phases from only the Fourier magnitudes, which is why it is called a phase retrieval problem.

A crystallized molecular structure may be represented by a 3-D periodic function, the so-called *electron density map* $\rho(\mathbf{r})$, where $\mathbf{r} = x\mathbf{a} + y\mathbf{b} + z\mathbf{c}$. The *fractional coordinates* x, y, and z take on values between 0 and 1. The *unit cell vectors* $\mathbf{a}$, $\mathbf{b}$, and $\mathbf{c}$ are neither necessarily orthogonal nor necessarily of equal length. Let $F(\mathbf{h^T})$ be the Fourier transform of $\rho(\mathbf{r})$, where $\mathbf{h} = h\mathbf{a}^* + k\mathbf{b}^* + l\mathbf{c}^*$. The *reciprocal lattice vectors* $\mathbf{a}^*$, $\mathbf{b}^*$, and $\mathbf{c}^*$ are defined such that $\mathbf{h^T r} = hx + ky + lz$ [116]. For simplicity, we confine the vectors $\mathbf{a}$, $\mathbf{b}$, and $\mathbf{c}$ to be three orthogonal vectors throughout this chapter. Thus, we think of $\mathbf{r}$ as simply a 3-D Cartesian coordinate (x, y, z); we write $\mathbf{r} = (x, y, z)$. Likewise, we think of $\mathbf{h}$ as a vector associated with three orthogonal vectors $\mathbf{a}^*$, $\mathbf{b}^*$, and $\mathbf{c}^*$, and simply write $\mathbf{h} = (h, k, l)$.

Since an electron density map is periodic, its associated Fourier transform consists of Dirac delta functions. These Dirac delta functions are regularly spaced along the reciprocal lattice vectors $\mathbf{a}^*$, $\mathbf{b}^*$, and $\mathbf{c}^*$. Therefore, we may think of h, k, and l as integers, and treat the Fourier transform as a discrete function indexed by $\mathbf{h}$. In x-ray crystallography, each value associated with a specific $\mathbf{h}$ is called a *structure factor*. Note that a structure factor consists of both Fourier magnitude and phase.

Since an electron density map and its associated structure factors form a Fourier transform pair, they can be written as

$$\begin{aligned} \rho(\mathbf{r}) &= \frac{1}{V} \sum_{\mathbf{h}} F(\mathbf{h^T}) \exp(-2\pi i \mathbf{h^T r}), \\ F(\mathbf{h^T}) = \mathscr{F}\{\rho\} &= \int_{\mathbf{r} \in \mathcal{R}} \rho(\mathbf{r}) \exp(2\pi i \mathbf{h^T r}) d\mathbf{r}, \end{aligned} \tag{5}$$

where V represents the volume of the unit cell, $\mathscr{F}$ represents the Fourier transform operation, the integral in Eq. (5) is three dimensional, and $\mathcal{R}$ represents the set of all the vectors in the unit cell. If the electron density map consists of atoms that may be approximated as "points" in space, we can think of F as the sum of each atom's scattering contribution. Then, F may be rewritten as

$$F(\mathbf{h^T}) = \sum_n f_n \exp(2\pi i \mathbf{h^T r}_n), \tag{6}$$

where f_n represents the scattering factor of the n-*th* atom [116]. Note that the retrieval of the Fourier phases from the Fourier magnitudes $|F|$ is equivalent to the reconstruction of ρ from $|F|$, since once we know the Fourier phases, we can take the inverse Fourier transform to obtain ρ.

2.2.2 The Space Groups

A *space group* is a combination of crystallographic symmetry operations with a *Bravais lattice* system. There are 14 Bravais lattices, such as primitive and body-centered, and five isometric [115] *symmetry operations*, which are the rotation axes, inversion axes, mirror planes, screw axes, and glide planes [9]. A Bravais lattice is discrete periodic array with an arrangement and orientation that appears exactly the same no matter which point of the array we view the array from. All "feasible" combinations of these lattice systems and symmetry operations, meaning those combinations that preserve periodicity and the lattice systems, lead to the 230 space groups.

A unit cell may have multiple molecules that are *equivalent* to each other under a space group. A single molecule among these multiple molecules is called an asymmetric unit. We can build up a complete unit cell by replicating an asymmetric unit according to the associated space group: $\rho(\mathbf{r}) = \sum_{j=1}^{J} \rho(\mathbf{G}_j\mathbf{r})$, where $\mathbf{r}$ belongs to an asymmetric unit (see Section 2.2.3 for relevant notation).

A plane group is similar in concept to a space group, except it is defined only in two dimensions. Some 2-D symmetry operations combined with 2-D plane lattice systems lead to the 17 plane groups [110]. The plane groups also often called the 2-D space groups. We will use some 2-D space group examples for our simulation study, but the general theory of our methods will be developed using the 3-D space groups.

2.2.3 Group Theory

An important property of a space group is that, as indicated by its name, it forms a "group" in a mathematical sense [115]. Let $\mathcal{SG}$ denote the set of all the 230 space groups. An element of this set is denoted by $\mathcal{G}_i$, where $i = 1, 2, \ldots, 230$. Each element of a space group $\mathcal{G}_i$ is denoted by $\mathbf{G}_j$, where $j = 1, 2, ..., J$, and J depends on which space group $\mathbf{G}_j$ belongs to. Each element of the space group can be represented by a matrix-vector pair $\mathbf{G}_j = (\mathbf{W}_j, \mathbf{w}_j)$, where $\mathbf{W}_j$ is a 3×3 matrix, and $\mathbf{w}_j$ is a 3-D column vector.

Now, we define $\mathbf{G}_j$'s operation on a coordinate to see how a space group actually forms a group.

Definition 1. (**Operation of Group Element**)

For a given coordinate $\mathbf{r}$, $\mathbf{G}_j\mathbf{r}$ *is defined by*

$$\mathbf{G}_j\mathbf{r} = (\mathbf{W}_j, \mathbf{w}_j)\mathbf{r} = \mathbf{W}_j\mathbf{r} + \mathbf{w}_j. \tag{7}$$

Cascaded applications of two space group elements result in:

$$\begin{aligned} \mathbf{r}' &= \mathbf{G}_k\mathbf{r} = (\mathbf{W}_k, \mathbf{w}_k)\mathbf{r} = \mathbf{W}_k\mathbf{r} + \mathbf{w}_k, \\ \mathbf{G}_j\mathbf{r}' &= (\mathbf{W}_j, \mathbf{w}_j)\mathbf{r}' = \mathbf{W}_j\mathbf{r}' + \mathbf{w}_j = \mathbf{W}_j\mathbf{W}_k\mathbf{r} + \mathbf{W}_j\mathbf{w}_k + \mathbf{w}_j. \end{aligned} \tag{8}$$

Hence, the composition of two space group elements can be reasonably defined as follows:

Definition 2. (**Composition of Group Elements**)

The group element composition $\mathbf{G}_l = \mathbf{G}_j\mathbf{G}_k$ *is defined by*

$$\mathbf{G}_l = \mathbf{G}_j\mathbf{G}_k = (\mathbf{W}_j, \mathbf{w}_j)(\mathbf{W}_k, \mathbf{w}_k) = (\mathbf{W}_j\mathbf{W}_k, \mathbf{W}_j\mathbf{w}_k + \mathbf{w}_j) = (\mathbf{W}_l, \mathbf{w}_l). \tag{9}$$

A space group with this group element composition forms a group in a mathematical sense if the conditions of an abstract group are satisfied [43, p. 6]. Wondratschek checks these conditions and shows that a space group is indeed a mathematical group [115]. A helpful property of a group is that the group composition $\mathbf{G}_l = \mathbf{G}_j\mathbf{G}_k$ in a group $\mathcal{G}_i$ is also in the group: $\mathbf{G}_l \in \mathcal{G}_i$. This property will be of theoretical importance in our methods.

2.2.4 Patterson Synthesis

A Patterson function is the autocorrelation function of ρ [91, p. 187]:

$$P(\mathbf{u}) = \int_{\mathbf{r}} \rho((\mathbf{r} + \mathbf{u}) \bmod \mathbf{d})\rho(\mathbf{r})d\mathbf{r}, \tag{10}$$

where $\mathbf{u} = (u, v, w)$ and $\mathbf{d} = (d_1, d_2, d_3)$. The vector elements d_1, d_2, and d_3 are the lengths of unit cell vectors $\mathbf{a}$, $\mathbf{b}$, and $\mathbf{c}$, respectively. For notational convenience, we confine $\mathbf{r}$ and $\mathbf{u}$ to those satisfying $(\mathbf{r} \bmod \mathbf{d}) = \mathbf{r}$, and $(\mathbf{u} \bmod \mathbf{d}) = \mathbf{u}$. Recall that $\mathbf{u}$ is associated with three orthogonal vectors, which actually coincide with the unit cell vectors, and hence, we write simply $\mathbf{u} = (u, v, w)$. Note that since ρ is periodic, the definition in Eq. (10) contains the modulo operation.

Since ρ is periodic, P is also periodic. In addition, P is an "aliased" autocorrelation. In some phase retrieval applications such as astronomical imaging [92], the function to be estimated has finite support, and its autocorrelation can be directly obtained in an "unaliased" form from photon differencing data. Comparing these two types of autocorrelation, we may better understand why phase retrieval problems in x-ray crystallography are more challenging than those in astronomical imaging.

P, the patterson function of ρ, can be directly obtained from the diffraction data $|F|^2$. From the convolution property of the Fourier transform, we obtain the following relation:

$$\mathscr{F}^{-1}\{|F|^2\} = P(\mathbf{u}) = \frac{1}{V}\sum_{\mathbf{h}} |F(\mathbf{h}^{\mathbf{T}})|^2 \exp(-2\pi i \mathbf{h}^{\mathbf{T}}\mathbf{u}). \tag{11}$$

This concept was first introduced in x-ray crystallography by Patterson [85, 86]. The Patterson function has been shown to be particularly useful when a molecular structure has a small number of heavy atoms [45] or the stereochemistry of a partial structure is known [30]. Using the Patterson function to find a molecular structure is called Patterson synthesis. Our methods fall into this category.

The symmetries of the Patterson functions are of special interest in our work. The symmetry of a Patterson function also forms a space group, namely one of the so-called *symmorphic groups*. A space group is called a symmorphic group when all the symmetry operations in the space group have one common point fixed [110]. Let $\mathcal{SSG}$ denote the set of all the symmorphic groups. $\mathcal{SSG}$ is a subset of $\mathcal{SG}$: $\mathcal{SSG} \subset \mathcal{SG}$. In 3-D, there are the 73 symmorphic groups among all the 230 space groups; in 2-D, there are the 13 symmorphic groups among all the 17 2-D space (or plane) groups. However, not all of them are the Patterson space groups. Let $\mathcal{PSG}$ be the set of all the possible Patterson space groups. Then, $\mathcal{PSG}$ has only 24 elements in 3-D and 7 elements in 2-D. Clearly, $\mathcal{PSG} \subset \mathcal{SSG}$. To distinguish these Patterson space groups from the other space groups, let $\mathcal{H}_m$ denote an element of $\mathcal{PSG}$, where $m = 1, 2, \ldots, 24$ (or 7).

Given an electron density map ρ, the Patterson space group associated with the space group of ρ can be deduced by two simple steps [110]. This property of the Patterson function will be given theoretical consideration in Section 2.4.1.

2.2.5 Csiszár's *I*-divergence

Csiszár's *I*-divergence is an information-theoretic discrepancy measure between two nonnegative functions [23]. This measure may be thought of as a generalization of the Kullback-Leibler distance [62]. However, unlike the Kullback-Leibler distance, Csiszár's *I*-divergence can accommodate cases involving two functions that have different integrals. For two nonnegative functions f and g, Csiszár's I-divergence is defined by

$$I(f||g) = \int \left\{ f(\mathbf{x}) \ln \frac{f(\mathbf{x})}{g(\mathbf{x})} - f(\mathbf{x}) + g(\mathbf{x}) \right\} d\mathbf{x}. \tag{12}$$

Minimizing Csiszár's *I*-divergence is the only estimation method that satisfies a certain set of postulates that is desirable for nonnegative linear inverse problems [23]. More practically, minimizing the *I*-divergence is asymptotically equivalent to maximum-likelihood estimation for a certain type of incomplete data problem [114]. Special cases include the Poisson intensity estimation problems from emission tomography [94,105] and astronomical imaging [92]. Snyder *et al.* discussed this important equivalence in their work on image deblurring subject to nonnegativity constraints [107].

2.2.6 The Schulz-Snyder Algorithm

The Schulz-Snyder algorithm is an iterative method for recovering nonnegative functions from their *n-th* order correlations [93]. Here, we focus specifically on the $n = 2$ case of recovery from autocorrelations, which is equivalent to phase retrieval. For execution on a computer, we discretize all functions of interest and make a slight abuse of notation by reusing symbols such as $\mathbf{x}$ to represent discretized coordinates. The algorithm estimates images from their autocorrelations by minimizing Csiszár's *I*-divergence:

$$I(S||R_f) = \sum_{\mathbf{y}} \left\{ S(\mathbf{y}) \ln \frac{S(\mathbf{y})}{R_f(\mathbf{y})} + R_f(\mathbf{y}) - S(\mathbf{y}) \right\}, \tag{13}$$

where $S = R_g$ is the autocorrelation of some true but unknown g that we want to estimate from S, and the autocorrelation of an estimate f is defined as

$$R_f(\mathbf{y}) = \sum_{\mathbf{x}} f(\mathbf{x}) f(\mathbf{x} + \mathbf{y}). \tag{14}$$

The algorithm aims to minimize the objective function $J(f) = I(S||R_f)$ subject to the constraints

$$
\begin{aligned}
C(f) = \sum_{\mathbf{x}} f(\mathbf{x}) &= C(g), \\
f &\geq 0,
\end{aligned}
\tag{15}
$$

where $[C(f)]^2 = \sum_{\mathbf{y}} S(\mathbf{y})$ (see Property 3.4 in [93, p. 1269]). Note that $C(g)$ can be obtained even if g is not known.

The Schulz-Snyder algorithm for recovering a nonnegative function from its autocorrelation is given by the iteration:

Algorithm 1.

$$
f_{k+1}(\mathbf{x}) = f_k(\mathbf{x}) \frac{1}{C(f)} \sum f_k(\mathbf{x}+\mathbf{y}) \frac{S(\mathbf{y})}{R_{f_k}(\mathbf{y})}. \tag{16}
$$

Note that if $f_0(\mathbf{x}) = 0$ for some particular $\mathbf{x}$, then $f_k(\mathbf{x}) = 0$ for that $\mathbf{x}$ for all k. This provides a convenient way of incorporating support constraints when they are available. This algorithm possesses some other nice properties such as monotonically decreasing I-divergence and conservation of total intensity of estimates, and its fixed points are minimizers of Eq. (13) [93].

2.3 The Schulz-Snyder Algorithm for X-ray Crystallography

The Schulz-Snyder algorithm (Algorithm 1) was originally designed for astronomical imaging, where we can obtain "unaliased" autocorrelations directly from the measurements [92]. Fortunately, if we replace the "unaliased" autocorrelation with the Patterson function, all the nice properties of Algorithm 1 remain, and hence, the algorithm can be also applied to x-ray crystallography with trivial modifications as shown in the derivation of Algorithm 2 in Appendix A.1. Since it is not difficult to show that the properties are still valid, we omit the proofs for conciseness. The arguments for the proofs are similar to those in Schulz and Snyder [93].

It is instructive to rewrite Algorithm 1 using the notation of x-ray crystallography:

Algorithm 2.

$$\rho_{k+1}(\mathbf{r}) = \rho_k(\mathbf{r})\frac{1}{C(\rho_k)}\sum_{\mathbf{u}}\rho_k((\mathbf{r}+\mathbf{u}) \bmod \mathbf{d})\frac{P(\mathbf{u})}{P_{\rho_k}(\mathbf{u})}, \tag{17}$$

where P denotes the Patterson function (obtained directly from the diffraction measurements) of some true electron density map ρ, P_{ρ_k} denotes the Patterson function of ρ_k (the *k-th* estimate of ρ), and $C(\rho_k)$ is given by

$$\begin{aligned} C(\rho_k) &= \sum_{\mathbf{r}}\rho_k(\mathbf{r}) \;=\; C(\rho),\ \forall k, \\ \rho &\geq 0, \end{aligned} \tag{18}$$

where $[C(\rho_k)]^2 = \sum_{\mathbf{u}} P(\mathbf{u})$. For computational purposes, all the functions involved are discretized, and hence, the Patterson function is redefined by

$$P(\mathbf{u}) = \sum_{\mathbf{r}}\rho((\mathbf{r}+\mathbf{u}) \bmod \mathbf{d})\rho(\mathbf{r}). \tag{19}$$

The Patterson functions of estimates are similarly defined. Note that Algorithm 2 still enjoys monotonically decreasing I-divergence.

Even though Algorithm 2 still preserves all the nice properties of Algorithm 1, there still may be some troublesome issues such as nonunique solutions, where there may exist two different electron density maps that produce the same Patterson function, and convergence to local minima that are not global minima, where the iterations can become trapped in "wrong" answers. Algorithm 1 also suffers from similar problems in Chapter 4, but these problems may be more serious in Algorithm 2.

2.4 Symmorphic-group Preservation Property of Algorithm 2

2.4.1 Patterson Space Groups

Since convolution in the spatial domain becomes multiplication in the Fourier domain, it may be easier to analyze the Patterson function in the Fourier domain. We start with a definition of the set of coordinates for an asymmetric unit.

Definition 3. (**Coordinate Set of an Asymmetric Unit**)

Let $\mathcal{I}(\mathcal{G}_i)$ denote the set of all the coordinates belonging to an asymmetric unit in an electron *density map ρ associated with the space group $\mathcal{G}_i$:*

$$\mathcal{I}(\mathcal{G}_i) = \{\mathbf{x} : \mathbf{x} \textit{ is in the asymmetric unit of } \rho\}. \tag{20}$$

Also, define the operation of a space group element on this set as follows:

$$\mathbf{G}_j\left(\mathcal{I}(\mathcal{G}_i)\right) = \{\mathbf{G}_j(\mathbf{x}) : \mathbf{x} \in \mathcal{I}(\mathcal{G}_i)\}, \tag{21}$$

where $j = 1, 2, \ldots, J$, $\mathbf{G}_j \in \mathcal{G}_i$, and $\mathbf{G}_1$ is defined as $(\mathbf{I}, \mathbf{0})$, which exists in any space group.

It follows from this definition that

$$\bigcup_{j=1}^{J} \mathbf{G}_j\left(\mathcal{I}(\mathcal{G}_i)\right) = \mathbf{U}, \ \bigcap_{j=1}^{J} \mathbf{G}_j\left(\mathcal{I}(\mathcal{G}_i)\right) = \emptyset, \tag{22}$$

where $\mathbf{U}$ is the set of all the coordinates in the unit cell. Similar notation can be also defined on the structure factors F when they possess symmetry.

The space group $\mathcal{G}_i$ of ρ results in another kind of symmetry in the associated structure factors:

$$\begin{aligned} F(\mathbf{h}^\mathbf{T}) &= \sum_n f_n \exp(2\pi i \mathbf{h}^\mathbf{T}\mathbf{r}_n) \\ &= \sum_n f_n \exp(2\pi i \mathbf{h}^\mathbf{T}\mathbf{G}_j\mathbf{r}_n) \\ &= \sum_n f_n \exp(2\pi i \mathbf{h}^\mathbf{T}(\mathbf{W}_j, \mathbf{w}_j)\mathbf{r}_n) \\ &= \sum_n f_n \exp(2\pi i \mathbf{h}^\mathbf{T}(\mathbf{W}_j\mathbf{r}_n + \mathbf{w}_j)) \\ &= \sum_n f_n \exp(2\pi i \mathbf{h}^\mathbf{T}\mathbf{W}_j\mathbf{r}_n)\exp(2\pi i \mathbf{h}^\mathbf{T}\mathbf{w}_j) \\ &= F(\mathbf{h}^\mathbf{T}\mathbf{W}_j)\exp(2\pi i \mathbf{h}^\mathbf{T}\mathbf{w}_j), \ \text{for all } \mathbf{G}_j \in \mathcal{G}_i, \end{aligned} \tag{23}$$

where the second equality holds since $\mathbf{G}_j\mathbf{r}_n$ is simply a rearrangement of all the $\mathbf{r}_n$. Then, it follows from Eq. (23) that

$$|F(\mathbf{h}^\mathbf{T})| = |F(\mathbf{h}^\mathbf{T}\mathbf{W}_j)|, \ \text{for all } \mathbf{G}_j \in \mathcal{G}_i. \tag{24}$$

Using Eq. (24), we can prove the following theorem:

Theorem 1. (**Patterson Space Group in 3-D**)
The Patterson space group $\mathcal{H}_m$ of P can be deduced from the space group $\mathcal{G}_i$ of ρ associated with P by applying the following two steps to $\mathcal{G}_i$:

(S1) If $\mathcal{G}_i$ has glide planes or screw axes, then they are replaced with the corresponding mirror planes and rotation axes, respectively. By doing so, we obtain a symmorphic group, say $\mathcal{G}_t$, which is not necessarily a Patterson space group yet.

(S2) If the symmorphic group $\mathcal{G}_t$ obtained in (S1) does not result in a centrosymmetric *Patterson function P, then the* inversions *of all the elements of $\mathcal{G}_t$ are added to form the Patterson space group $\mathcal{H}_m$ if they are not already in $\mathcal{G}_t$.*

Proof. See Appendix A.2. Part of the proof has been inspired by Shmueli [96]. □

The inversion of an element of a symmorphic group is defined by $(\mathbf{W}_j\mathbf{J}, \mathbf{0})$ where $\mathbf{J}$ is the 3×3 diagonal matrix whose elements are all -1. When the $\mathbf{J}$ operates on a point $\mathbf{u}$ in space, it results in another point $\mathbf{u}'$ that is symmetric to the point $\mathbf{u}$ about the origin. A function is called centrosymmetric if every point of the function has a corresponding symmetric point about the origin, namely $f(\mathbf{x}) = f(\mathbf{Jx})$. The Patterson space groups corresponding to each space group can be found in Hahn [110].

We obtain a corollary in 2-D:

Corollary 1. (**Patterson Plane Group in 2-D**)
The Patterson plane group $\mathcal{H}_m$ of P can be deduced from the plane group $\mathcal{G}_i$ of a 2-D ρ associated with P by applying the following two steps to $\mathcal{G}_i$:

(S1) If $\mathcal{G}_i$ has glide axes, then they are replaced with the corresponding reflection axes. By doing so, we obtain a 2-D symmorphic group $\mathcal{G}_t$, which is not necessarily a Patterson plane group yet.

(S2) If the symmorphic group $\mathcal{G}_t$ obtained in (S1) does not result in a centrosymmetric *Patterson function P, then the* inversions *of all the elements of $\mathcal{G}_t$ are added to form the Patterson space group $\mathcal{H}_m$ if they are not already in $\mathcal{G}_t$.*

Proof. The proof is similar to that of Appendix A.2. We omit this proof for conciseness. □

Planar centrosymmetry and inversions can be defined in the same way as in Theorem 1. However, the matrix $\mathbf{J}$ needs to be replaced with the 2×2 diagonal matrix whose elements are all -1.

2.4.2 Space-group Preservation Condition of Algorithm 2

Solutions to phase retrieval problems may be improved by incorporating more information into the solutions. If some symmetry properties of the function of interest are known, such as space group symmetries, these may give powerful information. [77]

We have seen that the correlation of two identical functions in Eq. (19) results in a special Patterson symmetry, which is another space group. Note that the summation in Algorithm 2 has a similar structure, in that it is a correlation between two functions in Algorithm 2 that have space groups. The difference between Algorithm 2 and the Patterson function is that the two functions have different space groups. This motivates a deeper investigation about the symmetry of the summation term at each iteration of Algorithm 2.

If the summation term in Algorithm 2 preserves the space group of ρ, then all the estimates produced by the algorithm will theoretically have the same space group if we initialize the algorithm with a starting point with that space group. When this is the case, it provides a convenient way for incorporating the known space group.

Let $Q_k(\mathbf{r})$ be the summation term in Algorithm 2:

$$\rho_{k+1}(\mathbf{x}) = \rho_k(\mathbf{x})\frac{1}{C(f)}Q_k(\mathbf{r}). \tag{25}$$

The constant term $1/C(f)$ is of no relevance to the space group of ρ_{k+1}. Let $\Lambda(\mathbf{h}^{\mathbf{T}})$ be the Fourier transform of $Q_k(\mathbf{r})$. Suppose that both the unknown ρ and the current estimate ρ_k have the same space group $\mathcal{G}_i$. The following properties hold from Eq. (23):

$$\begin{aligned} F(\mathbf{h}^{\mathbf{T}}) &= F(\mathbf{h}^{\mathbf{T}}\mathbf{W}_j)\exp(2\pi i\mathbf{h}^{\mathbf{T}}\mathbf{w}_j), \text{ for all } \mathbf{G}_j \in \mathcal{G}_i, \\ F_{\rho_k}(\mathbf{h}^{\mathbf{T}}) &= F_{\rho_k}(\mathbf{h}^{\mathbf{T}}\mathbf{W}_j)\exp(2\pi i\mathbf{h}^{\mathbf{T}}\mathbf{w}_j), \text{ for all } \mathbf{G}_j \in \mathcal{G}_i, \end{aligned} \tag{26}$$

so we obtain

$$\begin{aligned} \mathscr{F}\{Q_k(\mathbf{r})\} &= \Lambda(\mathbf{h}^{\mathbf{T}}) = \mathscr{F}\left\{\tilde{\rho}_k * \frac{P}{P_{\rho_k}}\right\} \\ &= F^*_{\rho_k}(\mathbf{h}^{\mathbf{T}})\frac{|F(\mathbf{h}^{\mathbf{T}})|^2}{|F_{\rho_k}(\mathbf{h}^{\mathbf{T}})|^2} = \frac{|F(\mathbf{h}^{\mathbf{T}})|^2}{F_{\rho_k}(\mathbf{h}^{\mathbf{T}})} \\ &= \frac{|F(\mathbf{h}^{\mathbf{T}}\mathbf{W}_j)|^2}{F(\mathbf{h}^{\mathbf{T}}\mathbf{W}_j)\exp(2\pi i\mathbf{h}^{\mathbf{T}}\mathbf{w}_j)} \\ &= \Lambda(\mathbf{h}^{\mathbf{T}}\mathbf{W}_j)\exp(-2\pi i\mathbf{h}^{\mathbf{T}}\mathbf{w}_j), \text{ for all } \mathbf{G}_j \in \mathcal{G}_i, \end{aligned} \tag{27}$$

where $\tilde{\rho}_k(\mathbf{r}) = \rho_k(-\mathbf{r})$, the operator $*$ denotes convolution, $F^*_{\rho_k}$ denotes the complex conjugate of F_{ρ_k}, and $\mathbf{G}_j = (\mathbf{W}_j, \mathbf{w}_j)$.

Using Eq. (27), we can prove the following theorem:

Theorem 2. (**Symmorphic-group Preservation of Algorithm 2**)
Let ρ have the space group $\mathcal{G}_i$. Also, let $\mathbf{G}_j \in \mathcal{G}_i$. Suppose Algorithm 2 is initialized with an image that has the space group $\mathcal{G}_i$, i.e., the space group of ρ_0 is $\mathcal{G}_i$. Then, the estimate ρ_k has the same space group for all $k = 0, 1, 2, \ldots$, if and only if the following condition is satisfied:

$$\mathbf{w}_j = \mathbf{0}, \text{ for all } \mathbf{G}_j \in \mathcal{G}_i. \tag{28}$$

Proof. See Appendix A.3. □

Note that the only space groups that can satisfy this condition are the symmorphic groups. (Recall that symmorphic groups are a superset of the Patterson space groups.)

An interesting corollary follows from this theorem:

Corollary 2. *Given an electron density map ρ with a space group $\mathcal{G}_i$:*

(i) Regardless of the kind of the space group $\mathcal{G}_i$ of ρ, $\rho * P_\rho$ and ρ have the same space group $\mathcal{G}_i$.

(ii) ρ and $\tilde{\rho} * P_\rho$ have the same space group $\mathcal{G}_i$ if and only if the space group $\mathcal{G}_i$ is a symmorphic group.

Proof. See Appendix A.4. □

2.5 Experiments

2.5.1 Sensitivity to Initial Estimates

In non-convex optimization problems, whether or not we will suffer from local minima depends on the choice of initial estimates. Global optimization techniques may partially avoid this problem; however, even with these techniques, there is still no guarantee that a global optimum will be attained in general.

It is often considerably difficult to characterize "good" initial estimates or visualize the objective-function surface in practice, especially in cases where the parameter space is large, such as in x-ray crystallography. For this reason, the best way to choose initial estimates remains elusive, unless some "helpful" prior information is available.

In Chapter 4, we present and discuss some numerical evidence that Csiszár's I-divergence surface in phase retrieval problems may have several local minima, meaning that the objective function is nonconvex even in cases where the image (and hence its autocorrelation) has finite support. We can reasonably argue that there may often be more local minima in x-ray crystallography due to the periodicity of image, which results in aliasing of the autocorrelation. Therefore, the choice of initial estimate has a significant impact on Algorithm 2.

Schulz and Snyder suggested initializing their iteration (Algorithm 1) with a constant plus a small amount of uniformly distributed random noise covering the known image support [93]. In the absence of support information, one can always try a constant-plus-noise rectangle at least half the length of the measured autocorrelation in each dimension, surrounded by zero padding. They add such noise to avoid having the algorithm enforce an unexpected symmetry on the estimates when the algorithm is initialized with a rectangular constant function or another function that is centrosymmetric about a point. This contrasts with some other applications of iterative algorithms, such as tomographic image reconstruction, where it is traditional to use a constant function as an initial estimate. However, a purely constant initial estimate should be completely avoided in Algorithm 2. Note that the Patterson function of a constant function is just a constant function; the shifted function of a constant is a constant function as well. Hence, in view of (17), we can easily see that if we initialize the algorithm with a constant function, then the algorithm simply multiplies the estimate by a constant at every iteration. It is easy to show that a constant function is a saddle point of Eq. (13) in the aliased case, and hence an undesirable fixed point of Algorithm 2.

The initial estimates suggested by Schulz and Snyder seemed to work reasonably well in their application, namely astronomical imaging [92, 93]. However, they did not provide a means for assessing performance of their initial estimates, nor is their choice guaranteed to work well in other applications. The general quantification of how well a specific initialization procedure may perform may not even be practically possible.

In fact, in many crystallographic reconstruction problems, the success of the algorithm is highly sensitive to the choice of initial estimate. Figure 1 illustrates an example of such sensitivity in crystallographic reconstruction, where the autocorrelation is aliased. Figure 1(a) shows the true image that we desire to estimate. The image consists of 65 by 65 pixels and has the space group $P2mm$, whose equivalent coordinates are (x, y), $(\overline{x}, \overline{y})$, $(\overline{x}, y)$, and $(x, \overline{y})$, where $\overline{x}$ is defined as $d_1 - x$, and d_1 is the dimension of the unit cell x axis. Figures 1(b), 1(c), and 1(d) show the final estimates obtained by applying the stopping criterion described below when the algorithm is initialized with a constant function added to images whose only nonzero pixels are located at $\{(5, 29), (5, 37), (61, 29), (61, 37)\}$, $\{(15, 19), (15, 47), (47, 19), (47, 47)\}$, and $\{(15, 2), (15, 64), (51, 2), (51, 64)\}$, respectively. The elements in each of these coordinate sets are four equivalent coordinates of the space group $P2mm$; each element has the row number and the column number of the corresponding greater-than-background pixel in the image. The constant function has the value 10, and the nonzero pixels were set to the same value 10 (yielding a total value of 20) for all three sets. As we can observe, the mere re-locations of the set of the four pixels produce different solutions, including a global minimum in Fig. 1(c); the convergence speeds are remarkably different as well. Since images shifted by an integer number of pixels (with wraparound in the periodic image case) and/or rotation by 180 degrees will produce the same Patterson (or autocorrelation) function, a global minimum may be shifted by an integer number of pixels and/or rotated by 180 degrees. Comparing Figs. 1(a) and 1(c) illustrates the wraparound shift effect. Some relevant data for this experiment are given in Table 1. The method $M2$, listed in this table, is described in Section 2.5.2.1.

In our experiments, we stopped running the algorithm at the first iteration when the maximum of the pixel differences between the current estimate and the previous estimate became less than 10^{-4}. We empirically settled on this stopping criterion for the purpose of this study. One reason we use this stopping criterion, instead of one based on the

Table 1: Selected data from the experiment associated with Figure 1. $I(P||P_{\rho_k})$ **represents the I-divergence value at the** k-th **iteration, and** k **is the number of iterations that were run to obtain the corresponding estimate.**

Figure	$I(P\|\|P_{\rho_0})$	$I(P\|\|P_{\rho_k})$	k	Method
1(b)	6.2987 $\times 10^8$	4.9987 $\times 10^4$	306007	$M2$
1(c)	2.7977 $\times 10^8$	157.2460	167062	$M2$
1(d)	5.3674 $\times 10^8$	2.7092 $\times 10^4$	194836	$M2$

I-divergence values, is that we have found that estimates may fluctuate even when the I-divergence is barely changing. Other researchers may prefer different stopping criteria for their application. In particular, the extreme number of iterations seen in Fig. 1 are intended to study the detailed behavior of the algorithm, and are probably overkill for everyday use. We will use this stopping criterion throughout our simulation studies, unless otherwise stated.

2.5.2 Reconstruction Examples

2.5.2.1 Preliminary Remarks

This section discusses some issues that arise in using Algorithm 2. While an extension to 3-D is readily available, we adhere to 2-D cases for simplicity and clarity of the illustrations and descriptions.

For simulation, we generated two different "truth" patterns. One is simple relative to the other in that it contains a smaller number of nonzero pixels. Note that, in general, just counting the number of nonzero pixels may not be the most instructive way to measure how simple patterns are. For this reason, we exaggerate the difference in simplicity of the two patterns by allocating a much larger number of nonzero pixels to the "relatively" complicated pattern. The size of all the patterns is 65 $\times$ 65; this implies that the size of the associated Patterson functions is also 65 $\times$ 65. We used $P2mm$ for the space group.

Algorithm 2 was applied to the patterns in three different ways, using the stopping criterion described above:

(M1) The algorithm was initialized with a random initial estimate, as suggested by Schulz and Snyder; no space group information is incorporated.

(M2) The algorithm is initialized with a random initial estimate that bears the known space group $P2mm$; however, the space group is not intentionally enforced on the estimates at each iteration. The space group center of the initial estimate is the origin. The asymmetric part of the initial estimate is generated as suggested by Schulz and Snyder.

(M3) The algorithm is initialized as in *M2*. Furthermore, the known space group is deliberately enforced on the estimates at every iteration by picking one of the asymmetric units and copying it according to the space group.

In spite of Theorem 2, the space group may not necessarily be perfectly preserved in practice due to the build up of numerical errors. This motivates *M3*. (As we will see later, it turns out that *M3* performs unexpectedly poorly. We will spend a considerable amount of time addressing this dramatic "plot twist.")

One purpose of this study is to show that *M2* can show improved performance over *M1*. However, since we initialize the algorithm randomly, and the algorithm is highly sensitive to the choice of initial estimates, it is not feasible to compare these two methods in a "perfectly fair" way. Therefore, we statistically assess the performance of the methods using numerous random initial estimates. Note that in x-ray crystallography, there are an enormous number of molecular structures, and these structures differ in complexity. Therefore, our conclusions in this simulation study may not be consistently true over all crystallographic structures. The performance of our methods will vary from structure to structure.

2.5.2.2 *Relatively Simple Pattern*

For some simple patterns, our algorithms are fairly successful whether or not the space group information is incorporated and regardless of how cleverly initial estimates are chosen. The three methods *M1*-*M3* were run 100 times with random initializations.

Figure 2(a) shows a simple pattern. The pattern contains about 200 nonzero pixels out of 65×65 pixels. Figures 3(a) and 3(b) show example random initial estimates with and without the known space group, respectively. Figures 3(c) and 3(d) show the Patterson functions of the initial estimates in Figs. 3(a) and 3(b), respectively. Figures 2(b), 2(c), and 2(d) show the final estimates produced by *M1*, *M2* and *M3*, respectively, where *M1* was initialized with the image in Fig. 3(a), and *M2* and *M3* were initialized with the image in Fig. 3(b). Figures 4(a), 4(b), 4(c), and 4(d) show the Patterson functions of the images in Figs. 2(a), 2(b), 2(c), and 2(d), respectively. To best show details on paper, the colormaps of the figures in this chapter are chosen such that the brightest pixel represents the lowest value, and the darkest pixel represents the highest value. The associated *I*-divergence values and the numbers of iterations are given in Table 2.

Table 2: Selected data from the experiment associated with Figure 2.

Figure	$I(P\|\|P_{\rho_0})$	$I(P\|\|P_{\rho_k})$	k	Method
2(b)	4.3619×10^8	8.6223	14761	$M1$
2(c)	8.2654×10^8	12.8607	8784	$M2$
2(d)	8.2654×10^8	4.3619×10^8	3586	$M3$

Statistically speaking, both *M1* and *M2* successfully reconstruct the simple pattern in Fig. 2(a). Both the methods succeeded in reconstructing "correct" solutions more than 90 times out of 100. Note that the correct solutions, global minima, are sometimes shifted and/or rotated versions of the original pattern in Fig. 2(a). *M2* was slightly more successful, but the difference was trivial: 93 successes with *M1*, compared with 97 successes with *M2*. However, *M3* hardly succeeded in reconstructing correct solutions; it was successful only four times in our 100 experiments. In those few times that *M3* was successful, the estimates had the correct space group with the space group center at the origin (as in the initial estimate). Hence, we conjecture that the initial estimates were luckily "good" enough to converge to correct answers. When we initialize *M1* and *M2* with these good initial estimates, we obtained estimates that were visually the same as those obtained by *M3*.

In comparing the results produced by *M2* and *M3*, an interesting argument can be formulated. In general, if the space group is forcibly constrained at each iteration, the algorithm is highly likely to converge to *symmorphic local minima.* By symmorphic local minima, we mean that if we add an arbitrarily small perturbation image that has the chosen space group and symmetry center to a symmorphic local minimum, the algorithm will pull the perturbed estimate back to the symmorphic local minimum. On the contrary, if the space group is merely encouraged by the choice of initial estiamte as in *M2*, instead of enforced as in *M3*, the algorithm tries to keep the space group (according to Theorem 2) until the point at which numerical errors accumulate to the point that the iteration leaves the space group. An intriguing aspect of this numerical aberration is that this temporary breaking of the space group may allow the algorithm to explore a new solution path, which may lead to one of the global minima, as shown in our experiment above. In these cases, the algorithm typically rediscovers and locks onto the correct space group again, except it picks a different point to center around than the center of the initial estimate. Note that *M3* converges to symmorphic local minima but *M2* converges to a global minimum even though the two methods were initialized with the same initial estimate; strangely, the numerical aberration helped the algorithm find a global minimum.

Figures 5, 6, and 7 show some interesting intermediate estimates obtained when $M1$, $M2$, and $M3$ are applied, respectively. Note that Figs. 6(a) and 7(b) are visually the same. In fact, $M3$ enforces the known space group, and hence, there are trivially small differences between the two images. Apart from this difference, the two images imply that for around 500 iterations, $M2$ and $M3$ produce almost the same estimates, provided that the methods are initialized with the same initial estimate with the known correct space group. However, $M2$ starts breaking the space group after the 500-*th* iteration, as shown in Fig. 6(b). This departure leads the algorithm to a global minimum by allowing the algorithm to explore a new path. On the contrary, $M3$ becomes trapped in a symmorphic local minimum as shown in Fig. 7.

Such phenomena appear to happen only when the estimates contain a large number of zeros. When we try to reconstruct images with the space group $P2mm$, but with no zeros, if the algorithm becomes trapped in symmorphic local minima, it does not seem to escape the symmorphic local minima via numerical quirks. Obviously, there exists some numerical errors over all the pixels. We conjecture that since the errors are relatively smaller than the estimate values, the errors are insufficient to break the space group. Meanwhile, when there are many values near zero in the estimates, the near-zero values become small enough that their relative differences due to the errors become more significant. These errors start breaking the equivalence of a few symmetric-set pixels in the beginning of the breaking stage, and such breaking causes some other breakings and so forth, *i.e.*, the errors accumulate faster and faster. Then, at some point, the disparity in the estimates becomes serious, and the algorithm reinterprets the space group center and tries to explore another estimate path. A similar numerical phenomenon was observed by Schulz and Snyder [93] in their unaliased case: "As can be seen from Eq. (3.15), if λ_k is symmetric about any point in $\mathcal{X}$, then λ_{k+1} will also be symmetric. Therefore, a uniform initial estimate will constrain the estimates to the set of symmetric images on $\mathcal{X}$. Although in many simulation experiments finite precision has permitted the images to leave this set, it is important not to rely on this uncertainty by selecting an asymmetric initial image."

Figure 8 depicts possible estimate paths that methods *M1*, *M2*, and *M3* might follow. S_k, for $k = 1, 2, \ldots, n$, are sets of estimates that have space group $P2mm$. The estimates in S_1 have the space group center at the origin; the other sets have the space group center at other points. All estimates in a particular set S_k have a common space group center. P1 and P2 are estimate paths that method *M1* may follow. I_1 and I_2 represent different initializations. One initialization may lead the algorithm to a global minimum (see G_1) or a local minimum (see L_1). A global minimum will be located close to or in a S_k since the estimate will have the known space group, perhaps slightly degraded by numerical errors. P3 and P4 are estimate paths that *M2* and *M3* (respectively) may follow when the common initial estimate I_4 is used. For these two paths, if numerical errors permit the algorithm to explore another path by breaking the initial space group/center, then the algorithm may find a global minimum (see G_2), which is close to S_n; however, if the space group is constrained, the algorithm may only produce a symmorphic local mimimum (see L_3). P5 and P6 are additional estimate paths that *M2* and *M3* may follow, respectively. Note that the initial estimate (I_3) is also common to *M2* and *M3*. However, in this case, the algorithm leads to only local minima (see L_4 and L_2), no matter which method is applied. P7 represents an estimate path that *M3* may follow when an initial estimate (I_5) is "good" enough to lead the algorithm to a global minimum (see G_3). Recall that this good estimate will also lead *M2* and *M3* to the same global minimum, within expected numerical tolerances.

2.5.2.3 More Complicated Patterns

As the pattern that we want to estimate becomes more complicated, it typically becomes more difficult for the algorithm to find a correct solution. Figure 9 shows a pattern that is more complicated than the pattern discussed in Section 2.5.2.2, in the sense that it contains more nonzero pixels. Again, in general, the number of nonzero pixels is not necessarily a good measure of complexity of a pattern; we are using a simplified example. Figure 9(a) shows the original pattern that we desire to estimate. Figures 9(b), 9(c), and 9(d) are estimates produced by *M1*, *M2*, and *M3*, respectively. Figures 10(a) and 10(b) show the initial estimates used in *M1*, and *M2* and *M3*, respectively. Figures 10(c) and 10(d) show the Patterson functions of Figs. 10(a) and 10(b), respectively. Figures 11(a), 11(b), 11(c), and 11(d) are the Patterson functions corresponding to Fig. 9(a), 9(b), 9(c), and 9(d), respectively. In contrast with the case in Section 2.5.2.2, *M1*, which starts with a nonsymmetric estimate, could reconstruct correct images only 6 times out of 100. However, *M2*, which starts with an estimate with the correct symmetry, reconstructed correct images 24 times out of 100, which may be regarded as an improvement. Recall that we initialize the methods with randomly generated images; hence, we can only compare these methods statistically. Like in Section 2.5.2.2, *M3* hardly ever succeeded in reconstructing a correct solution. In fact, in this experiment, *M3 never* converged to a global minimum in 100 tries.

This experiment illustrates all the various estimate paths discussed in Section 2.5.2.2. Figures 12, 13, and 14 show some interesting intermediate estimates. Figure 13(b) shows the numerically-induced departure from the initial space group/center combination; this path eventually leads to a global minimum. Again note that the estimates in Figs. 13(a) and 14(b) are almost the same, which means *M2* and *M3* follow the same path until the iteration. In this case as well, breaking off from the initial space group allows the algorithm to follow an alternative path, leading to a global minimum as seen in Fig. 13. However, with *M3*, the algorithm becomes trapped in a symmorphic local minimum because of the enforced space group. The *I*-divergence values and the number of iterations associated with one of the 100 runs of this experiment are given in Table 3.

Table 3: Selected data from the experiment associated with Figure 9.

Figure	$I(P\|\|P_{\rho_0})$	$I(P\|\|P_{\rho_k})$	k	Method
9(b)	2.9497 $\times 10^8$	2.6541 $\times 10^4$	204274	$M1$
9(c)	2.3292 $\times 10^9$	34.9219	109289	$M2$
9(d)	2.3292 $\times 10^9$	1.3557 $\times 10^5$	13792	$M3$

2.5.2.4 Effects of M2

We have hinted that *M2* statistically outperforms *M1* with respect to what percentage of our runs converged to the desired result. The following example further explores this idea. In this experiment, we initialize Algorithm 2 with two simple initial estimates made by adding a constant (5 in each initial estimate) and an image whose only nonzero pixels are located at {(16,18), (16,48),(50,18),(50,48)}. In one initial estimate, all four nonzero pixels are set to different values, namely 6, 11, 15, and 30. In the other, all these pixels are set to the same value of 10. Hence, the latter has the known space group, but the former does not; these initializations correspond to *M2* and *M1*, respectively.

Figures 15(a) and 15(c) show the two initial estimates, and Figs. 15(b) and 15(d) show their Patterson functions, respectively. Figures 16(a) and 16(c) show the final estimates produced by Algorithm 2 when the algorithm is initialized with the images in Figs. 15(a) and 15(b), respectively. The associated *I*-divergence values and the numbers of iterations are given in Table 4. The simple difference between the two initial estimates leads the algorithm to totally different results. One converges to a global minimum, but the other does not. This experiment suggests that the incorporation of a known space group may help the algorithm find a better solution in some cases.

Table 4: Selected data from the experiment associated with Figure 16.

Figure	$I(P\|\|P_{\rho_0})$	$I(P\|\|P_{\rho_k})$	k	Method
16(a)	2.9576 $\times 10^8$	2.6521 $\times 10^4$	307332	$M1$
16(c)	2.9492 $\times 10^8$	134.1945	179594	$M2$

2.6 Conclusions

Intrigued by the prospect of using minimum I-divergence techniques in x-ray crystallography, we have created a slight modification of the Schulz-Snyder algorithm as a first step in that direction. Furthermore, we proved that our modified Schulz-Snyder algorithm *theoretically* preserves known symmorphic space group structures. This is useful since a crystal's space group can be easily extracted directly from x-ray diffraction data. Exploiting this property provides a way to incorporate this space group information in the symmorphic case.

Since the algorithm is so sensitive to the choice of initial estimate, it is hard to study the effect of incorporating space group information in the initial estimate. For this reason, our comparison was conducted based on a statistical assessment. In some "simple" cases, both methods often converge to global minima. Statistically speaking, incorporating the correct space group in the initial estimate yields advantages over not doing so, both in terms of converging to a global minima and the number of iterations needed to get there.

One astonishing observation is that in practice, unless the known space group is deliberately enforced at each iteration, numerical errors may eventually cause the algorithm to slip off of the space group; but such a departure sometimes serendipitously leads to a global minimum. Examples of the gap between theory and practice being to our advantage are rare, but this seems to be one of them!

The most meaningful avenue for future research would be to characterize a good initial estimate for the algorithms.

This study is entirely devoted to the case of symmorphic space groups. Nonsymmorphic space groups also may be incorporated into Algorithm 2. Unlike the symmorphic space groups, nonsymmorphic space groups need to be deliberately enforced at every iteration. However, such enforcement would probably lock up the algorithm on local minima, as our experiments with method *M3* may imply. One alternative to the strict "copy and paste" symmetry enforcement might be "soft" enforcement via a penalty added to the objective function. One could form a penalty based on the I-divergence between the corresponding points of the different asymmetric units. By tweaking a constant multiplying the penalty, the symmetry could be enforced with varying degrees of strictness as the iterations progress. This approach could be employed for both symmorphic and nonsymmorphic space groups.

This study assumed that the data are not corrupted by noise, which is never the case in practice. In Chapter 5, we study the effect of noise in minimum I-divergence phase retrieval for both the aliased and unaliased cases for non-symmetric images. The study of the interaction of noise and symmetry remains a topic for future research.

This chapter focused on the case where the lattice vectors **a**, **b**, and **c** are orthogonal. Our algorithms would require some modifications to work for the nonorthogonal cases. It is not clear at present how easy or difficult such modifications will be, or what theoretical symmetry-preserving properties, if any, they might possess.

For computational convenience and ease of exposition, we conducted our experiments in 2-D. Of course, real crystallographic structures are 3-D, yielding additional computational complexity in two ways: 1) each individual iteration takes more time, and 2) as the number of parameters increases, we have found that our algorithms sometimes take longer to converge (as is often seen in EM algorithms).

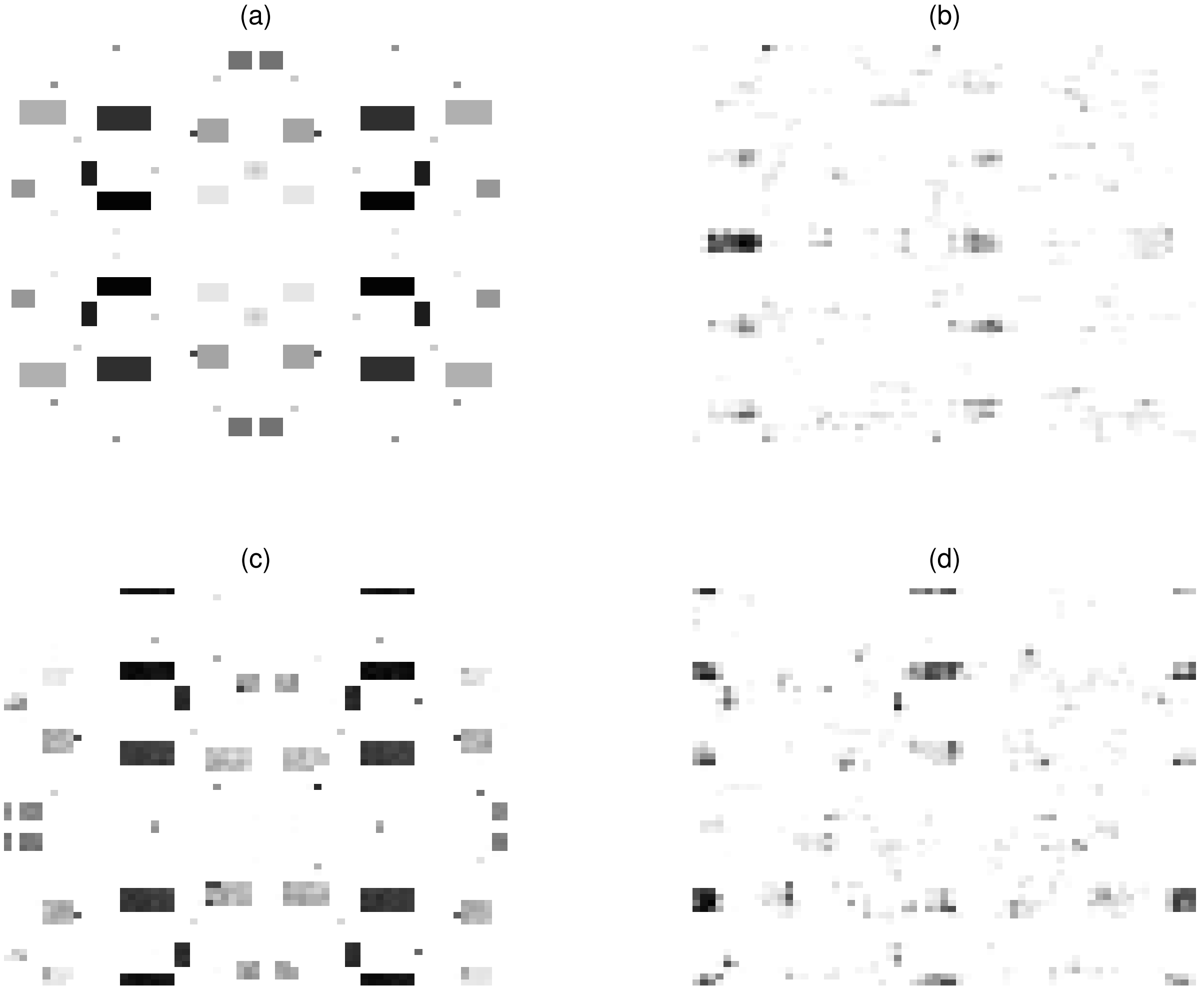


Figure 1: This illustration shows an example of the sensitivity of Algorithm 2 to the choice of initial estimate. The algorithm is initialized with a constant function 10 added to a function whose only nonzero pixels are located at the given *location sets*. All these nonzero pixels have the same value 10. (a) Original true image. (b) Final estimate obtained when the location set is $\{(5,29),(5,37),(61,29),(61,37)\}$. (c) Final estimate obtained when the location set is $\{(15,19),(15,47),(47,19),(47,47)\}$. (d) Final estimate obtained when the location set is $\{(15,2),(15,64),(51,2),(51,64)\}$.

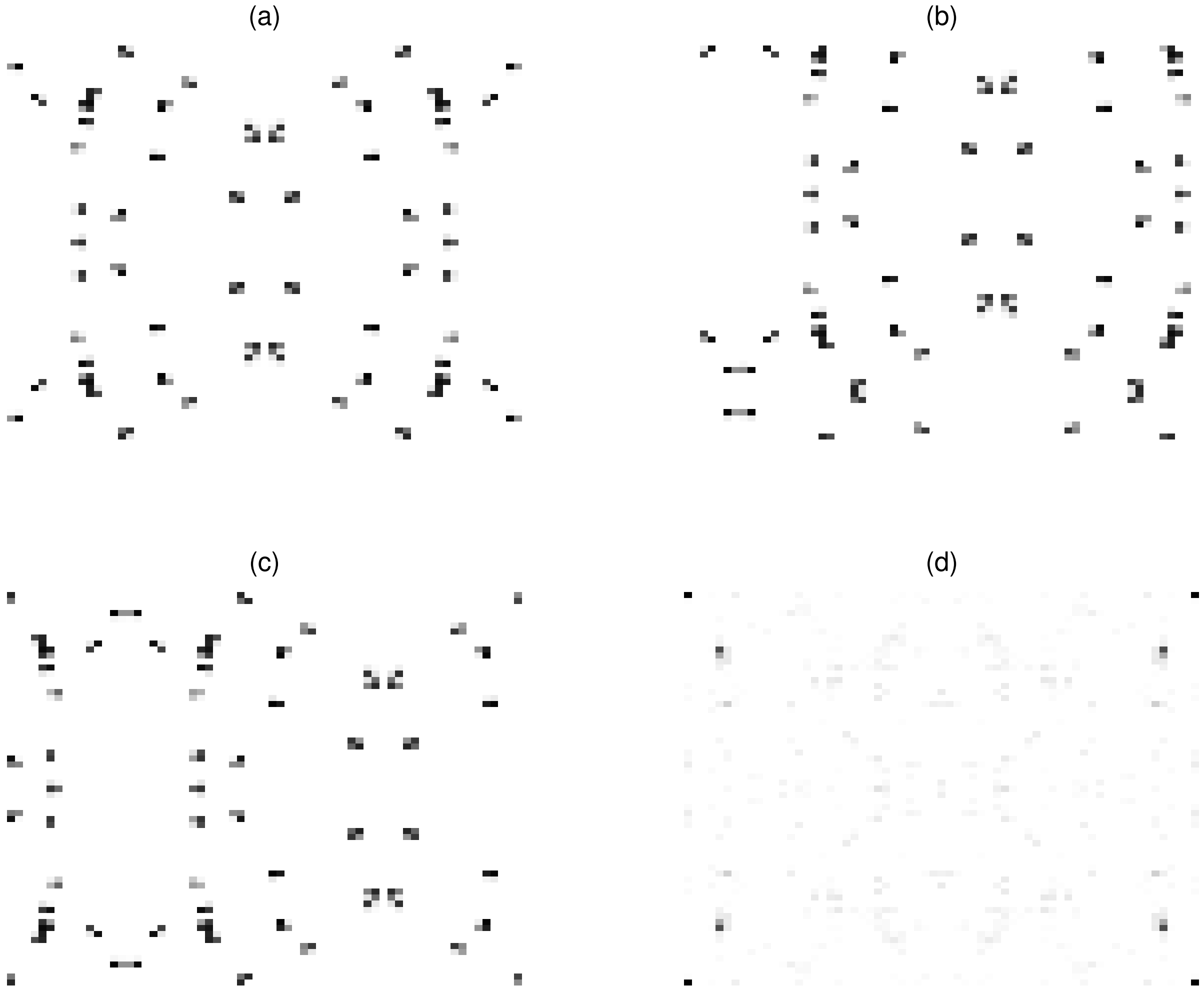


Figure 2: (a) Original true image consisting of 200 nonzero pixels out of 65 × 65 pixels. (b) Final estimate when *M1* is applied; the initial estimate is given in Fig. 3(a). (c) Final estimate when *M2* is applied; the initial estimate is given in Fig. 3(c). (d) Final estimate when *M3* is applied; the initial estimate is given in Fig. 3(c).

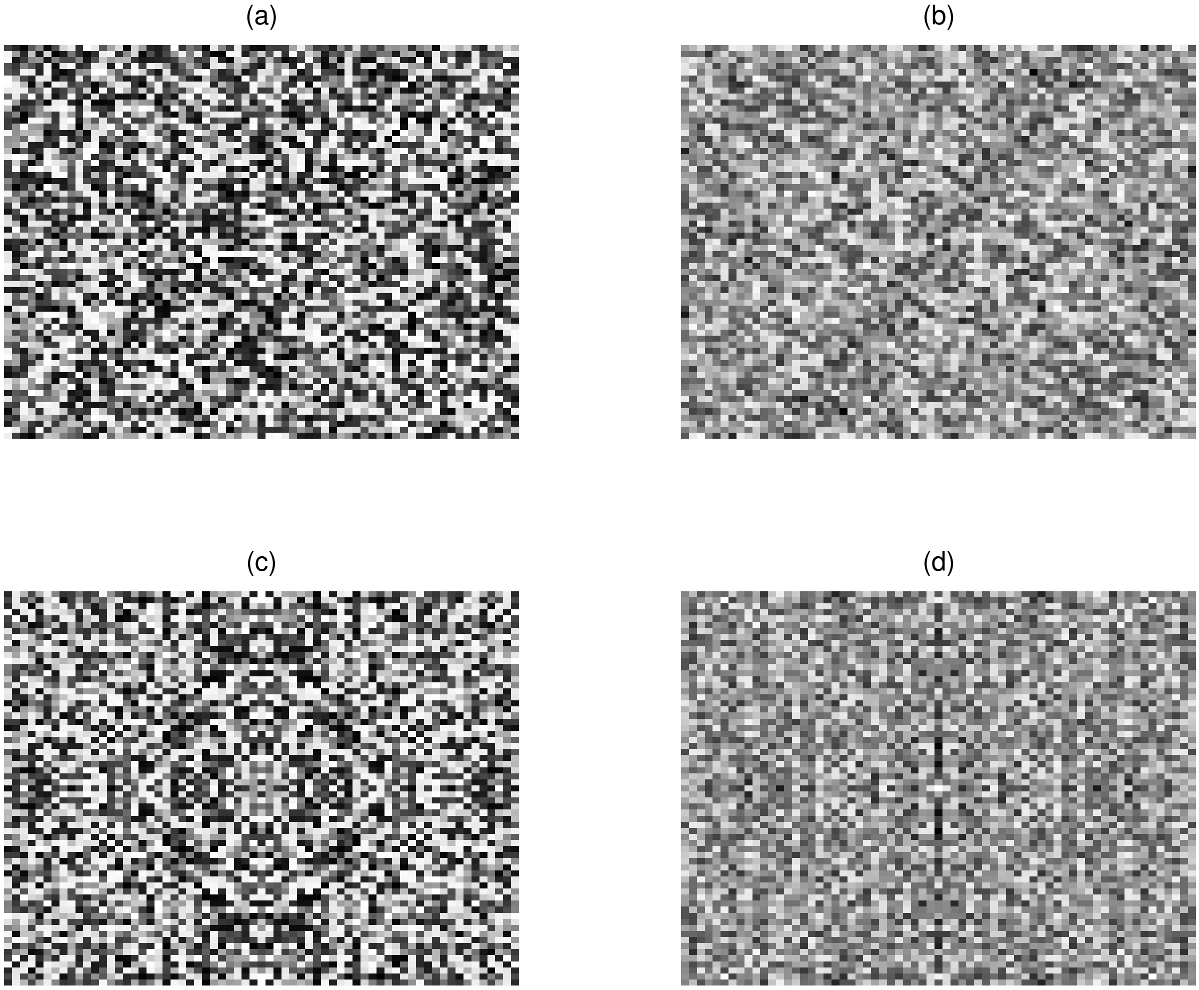


Figure 3: (a) Initial estimate generated as Schulz and Snyder suggested; the estimate does not have the space group $P2mm$. (b) Patterson function of the initial estimate given in (a). (c) Initial estimate that has the space group $P2mm$; the asymmetric part in this estimate was generated as Schulz and Snyder suggested. (d) Patterson function of the initial estimate given in (c). (Note: To best show detail, the large peaks of the autocorrelations in Figs. 3(b) and 3(d) were removed, and the autocorrelations are shown with a logarithmic scale.)

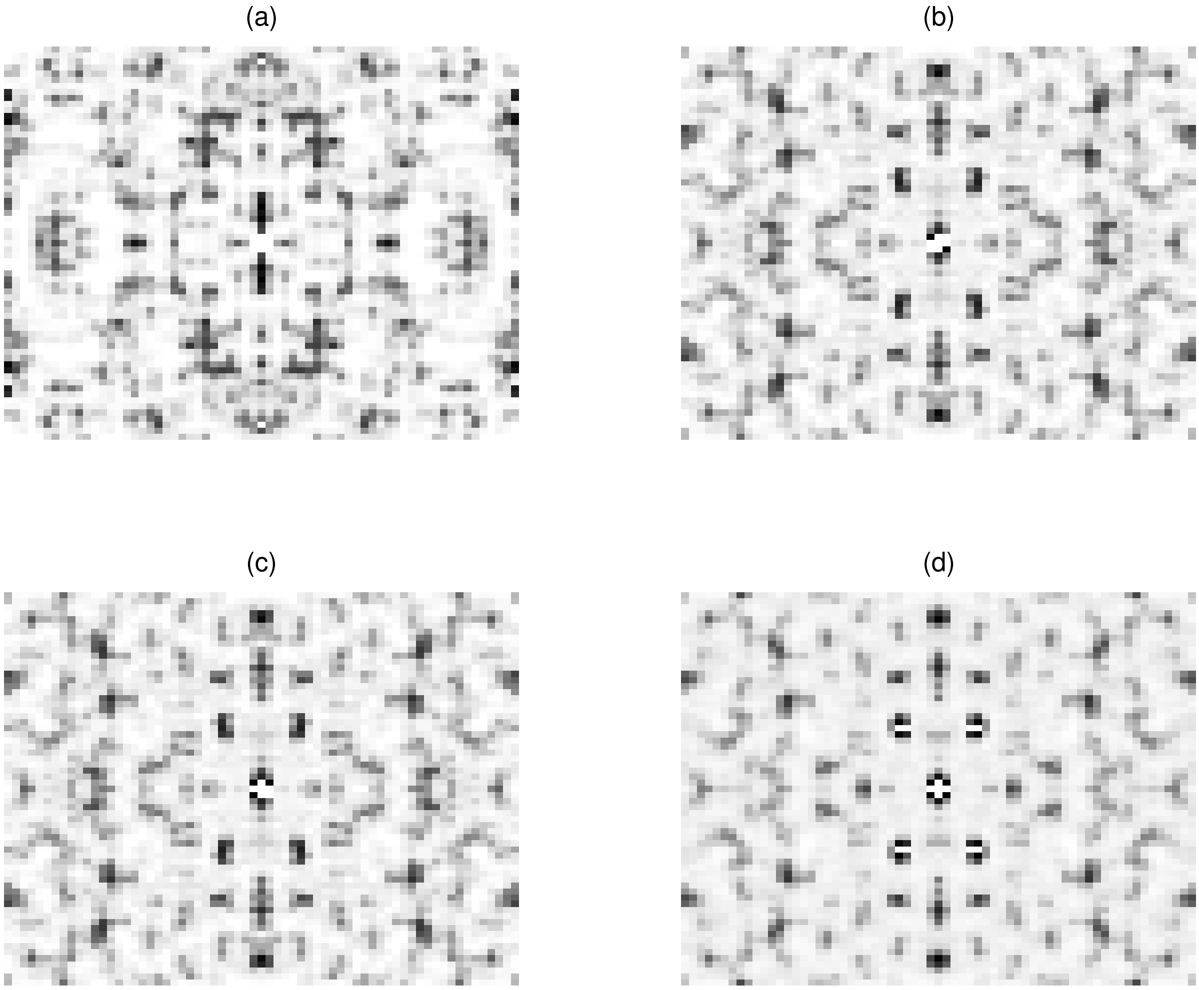


Figure 4: (a) Patterson function of Fig. 2(a). (b) Patterson function of Fig. 2(b) (c) Patterson function of Fig. 2(c). (d) Patterson function of Fig. 2(d). (Note: The large peaks of the autocorrelations were removed to best show detail.)

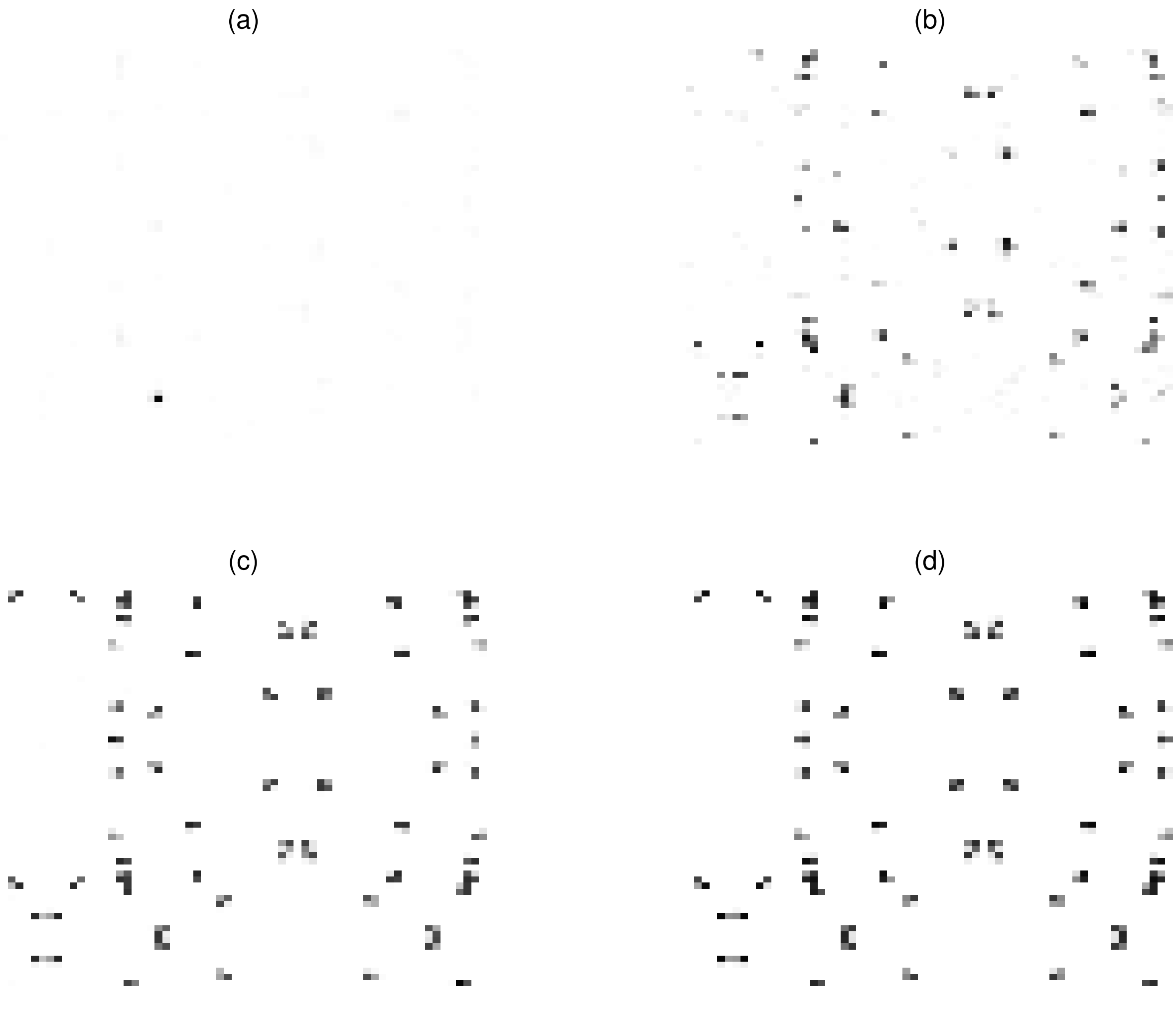


Figure 5: Some interesting intermediate estimates of the pattern in Fig. 2(a) provided by the algorithm when $M1$ is applied: (a) Estimate at the 200-th iteration. (b) Estimate at the 1300-th iteration. (c) Estimate at the 2400-th iteration. (d) Estimate at the 14000-th iteration.

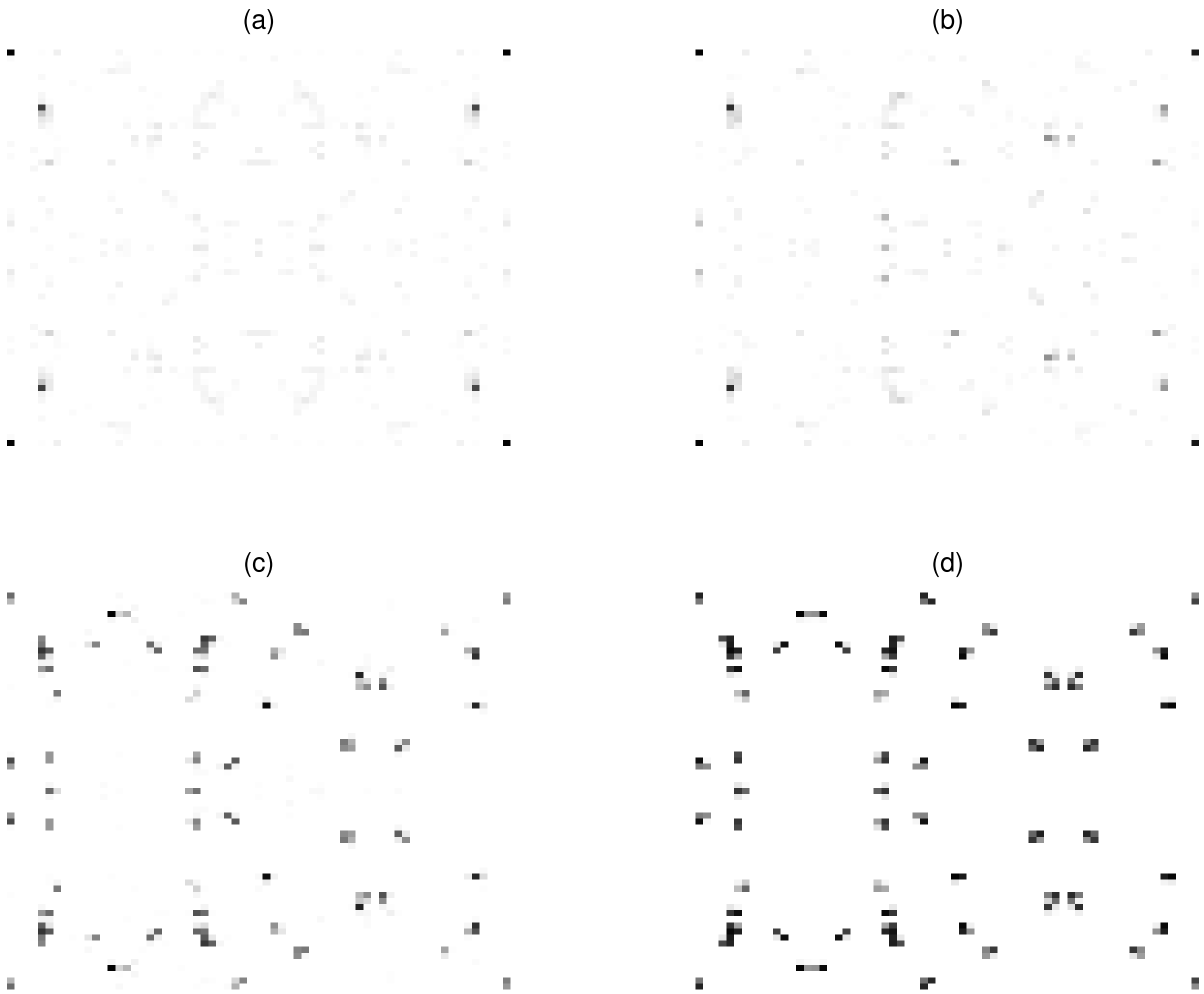


Figure 6: Some interesting intermediate estimates of the pattern in Fig. 2(a) provided by the algorithm when $M2$ is applied: (a) Estimate at the 500-th iteration. (b) Estimate at the 600-th iteration. (c) Estimate at the 800-th iteration. (d) Estimate at the 6000-th iteration.

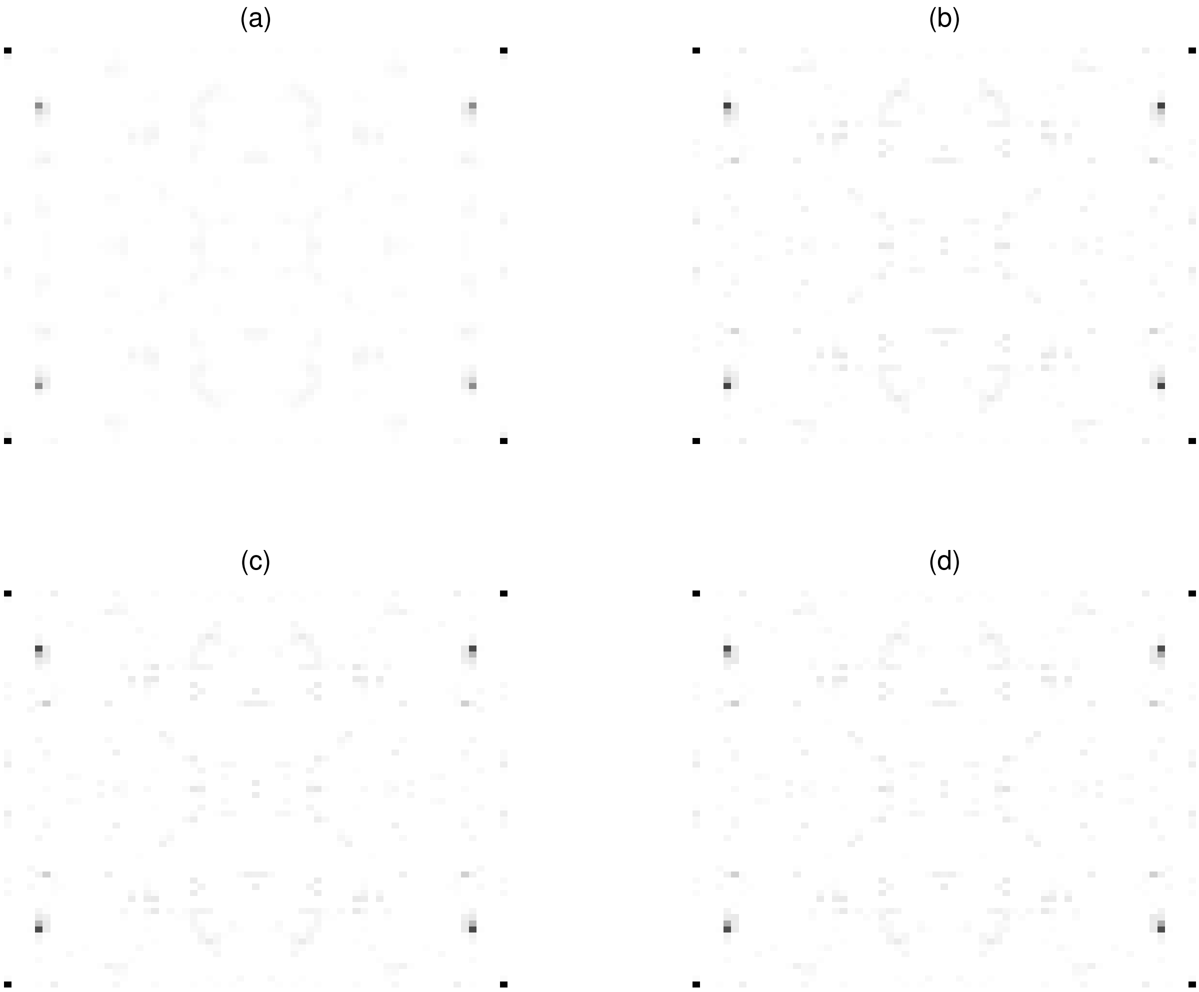


Figure 7: Some interesting intermediate estimates of the pattern in Fig. 2(a) provided by the algorithm when $M3$ is applied: (a) Estimate at the 200-th iteration. (b) Estimate at the 500-th iteration. (c) Estimate at the 800-th iteration. (d) Estimate at the 2800-th iteration.

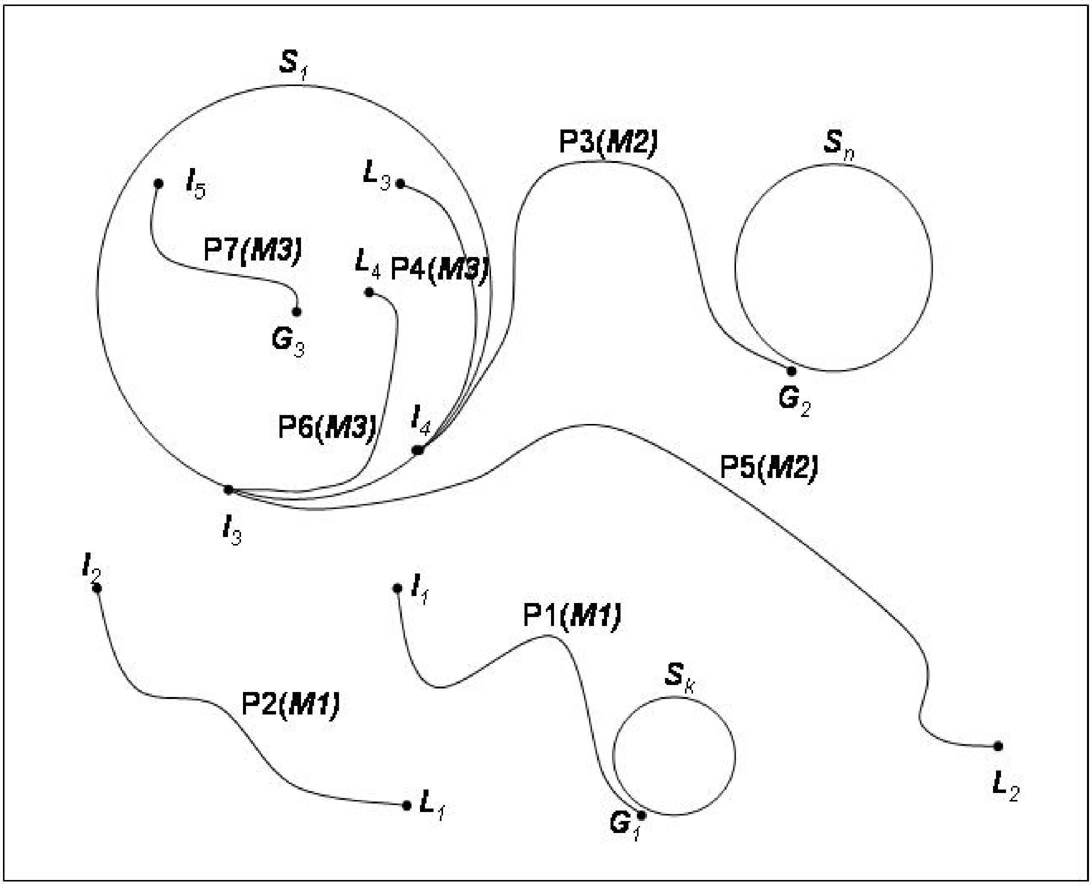


Figure 8: Illustration of possible estimate paths that our methods follow. S_k represents a set of estimates that have the known space group, I_k an initial estimate, L_k and G_k a local and global minimum, respectively, Pk an estimate path, and Mk a method that produces the associated estimate path.

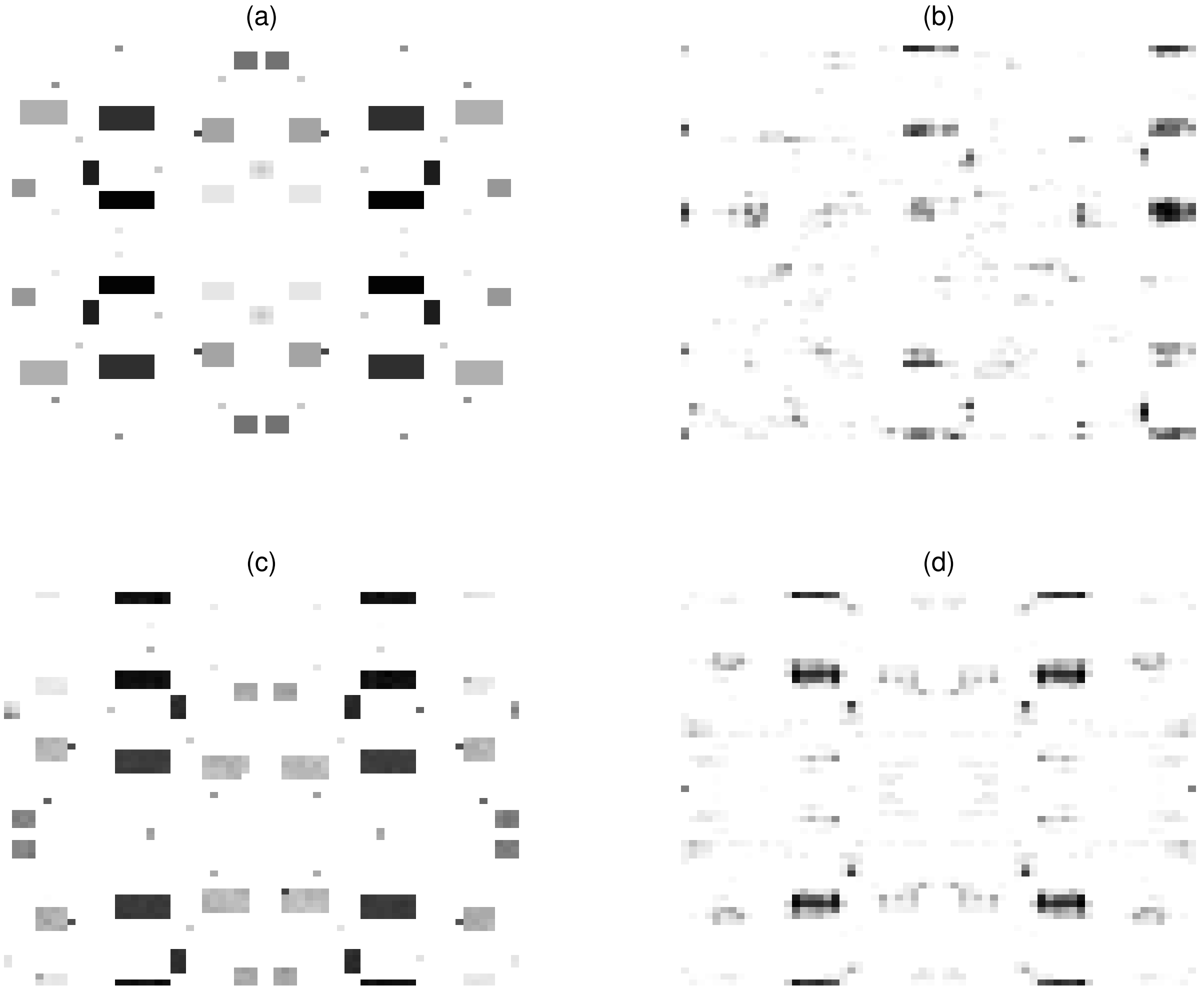


Figure 9: (a) Original true image consisting of 558 nonzero pixels out of 65 × 65 pixels. (b) Final estimate when *M1* is applied; the initial estimate is given in Fig. 10(a). (c) Final estimate when *M2* is applied; the initial estimate is given in Fig. 10(c). (d) Final estimate when *M3* is applied; the initial estimate is given in Fig. 10(c).

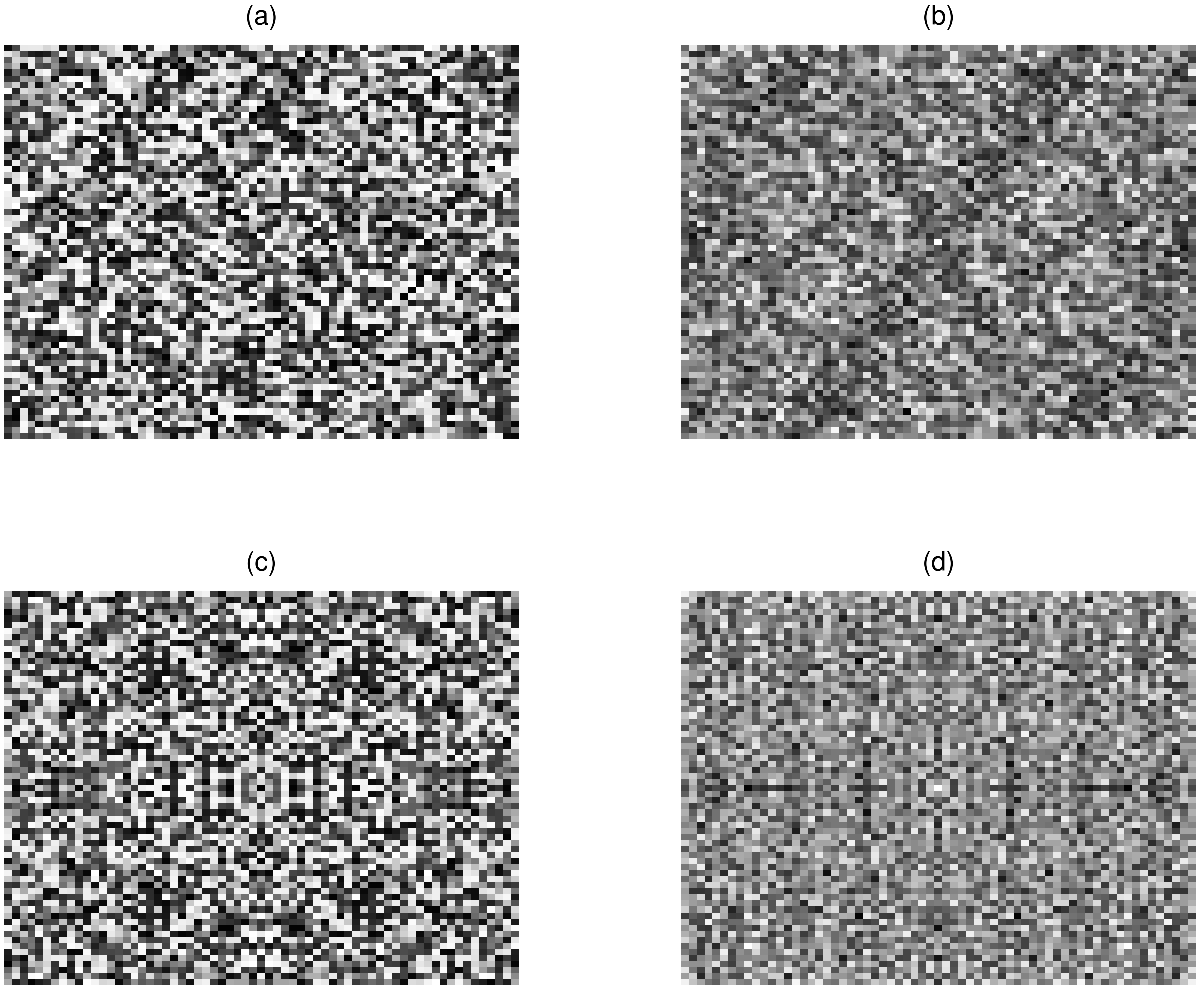


Figure 10: (a) Initial estimate generated as Schulz and Snyder suggested; the estimate does not have the space group $P2mm$. (b) Patterson function of the initial estimate given in (a). (c) Initial estimate that has the space group $P2mm$; the asymmetric part in this estimate was generated as Schulz and Snyder suggested.(d) Patterson function of the initial estimate given in (c). (Note: To best show detail, the large peaks of the autocorrelations in Figs. 10(b) and 10(d) were removed, and the autocorrelations are shown with a logarithmic scale.)

Figure 11: (a) Patterson function of Fig. 9(a). (b) Patterson function of Fig. 9(b) (c) Patterson function of Fig. 9(c). (d) Patterson function of Fig. 9(d).

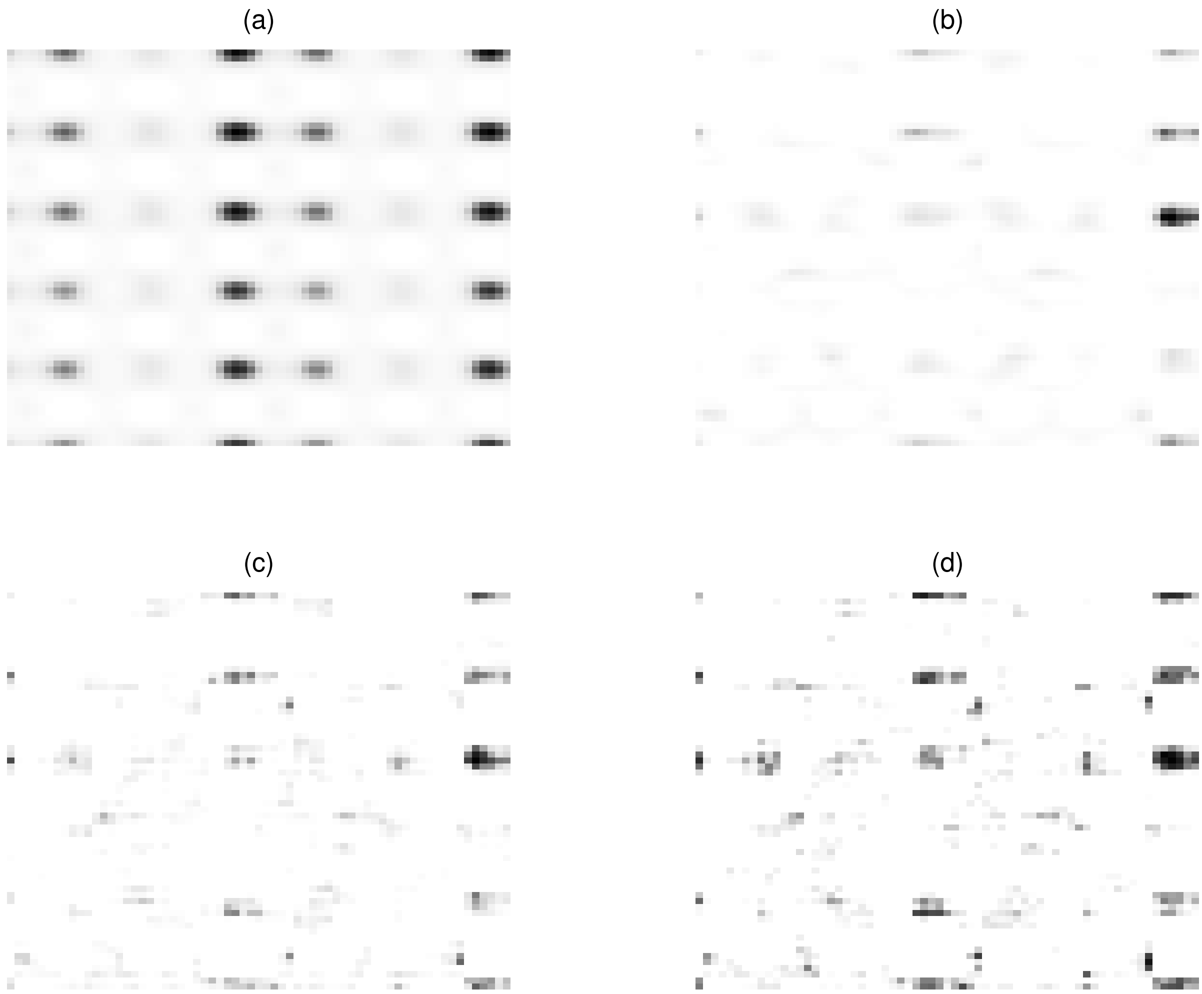


Figure 12: Some interesting intermediate estimates of the pattern in Fig. 9(a) provided by the algorithm when $M1$ is applied: (a) Estimate at the 100-th iteration. (b) Estimate at the 300-th iteration. (c) Estimate at the 1700-th iteration. (d) Estimate at the 200000-th iteration.

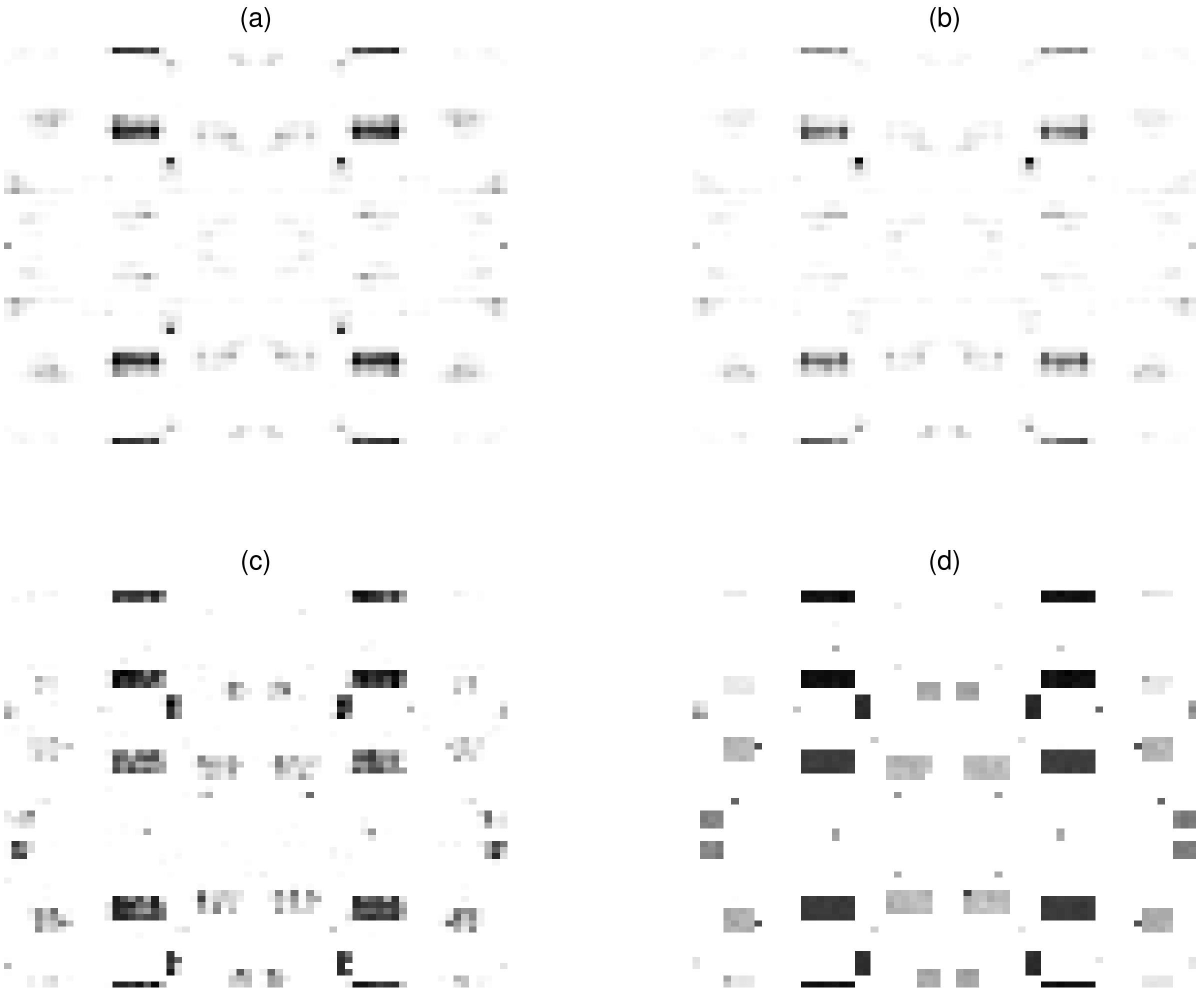


Figure 13: Some interesting intermediate estimates of the pattern in Fig. 9(a) provided by the algorithm when $M2$ is applied: (a) Estimate at the 1100-*th* iteration. (b) Estimate at the 1500-*th* iteration. (c) Estimate at the 20000-*th* iteration. (d) Estimate at the 100000-*th* iteration.

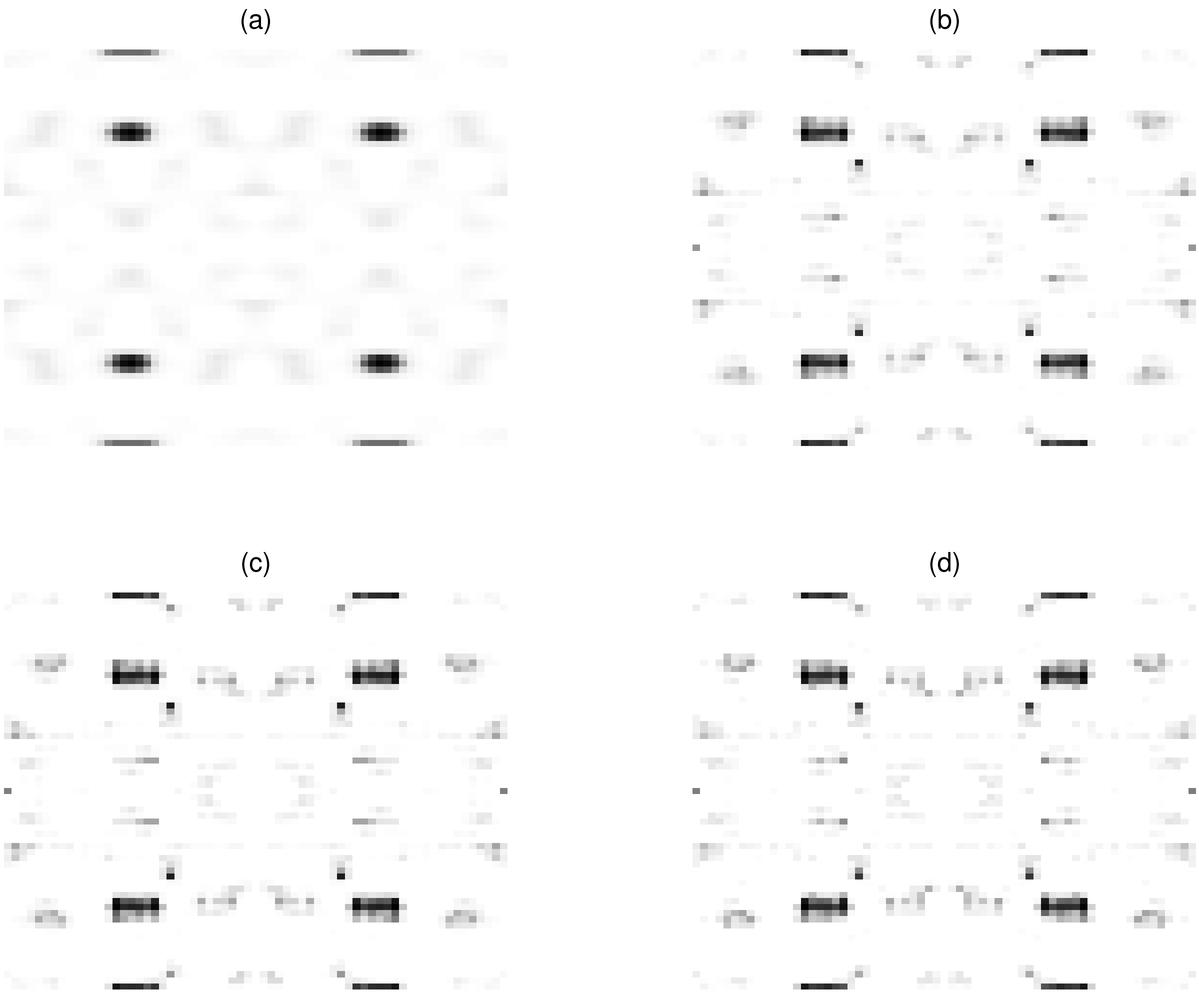


Figure 14: Some interesting intermediate estimates of the pattern in Fig. 9(a) provided by the algorithm when $M3$ is applied: (a) Estimate at the 300-*th* iteration. (b) Estimate at the 1100-*th* iteration. (c) Estimate at the 1500-*th* iteration. (d) Estimate at the 13000-*th* iteration.

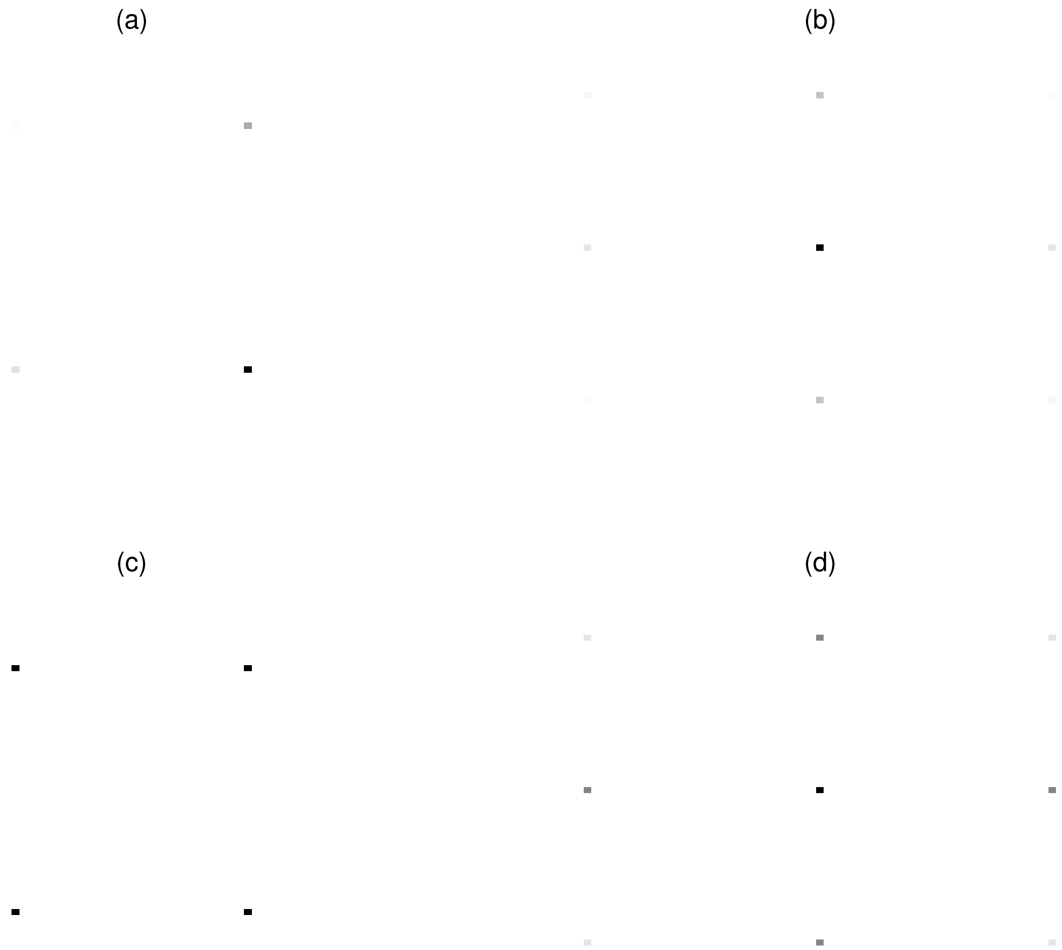


Figure 15: (a) Initial estimate with no space group. (b) Patterson function of the initial estimate given in (a). (c) Initial estimate with space group $P2mm$. (d) Patterson function of the initial estimate given in (c).

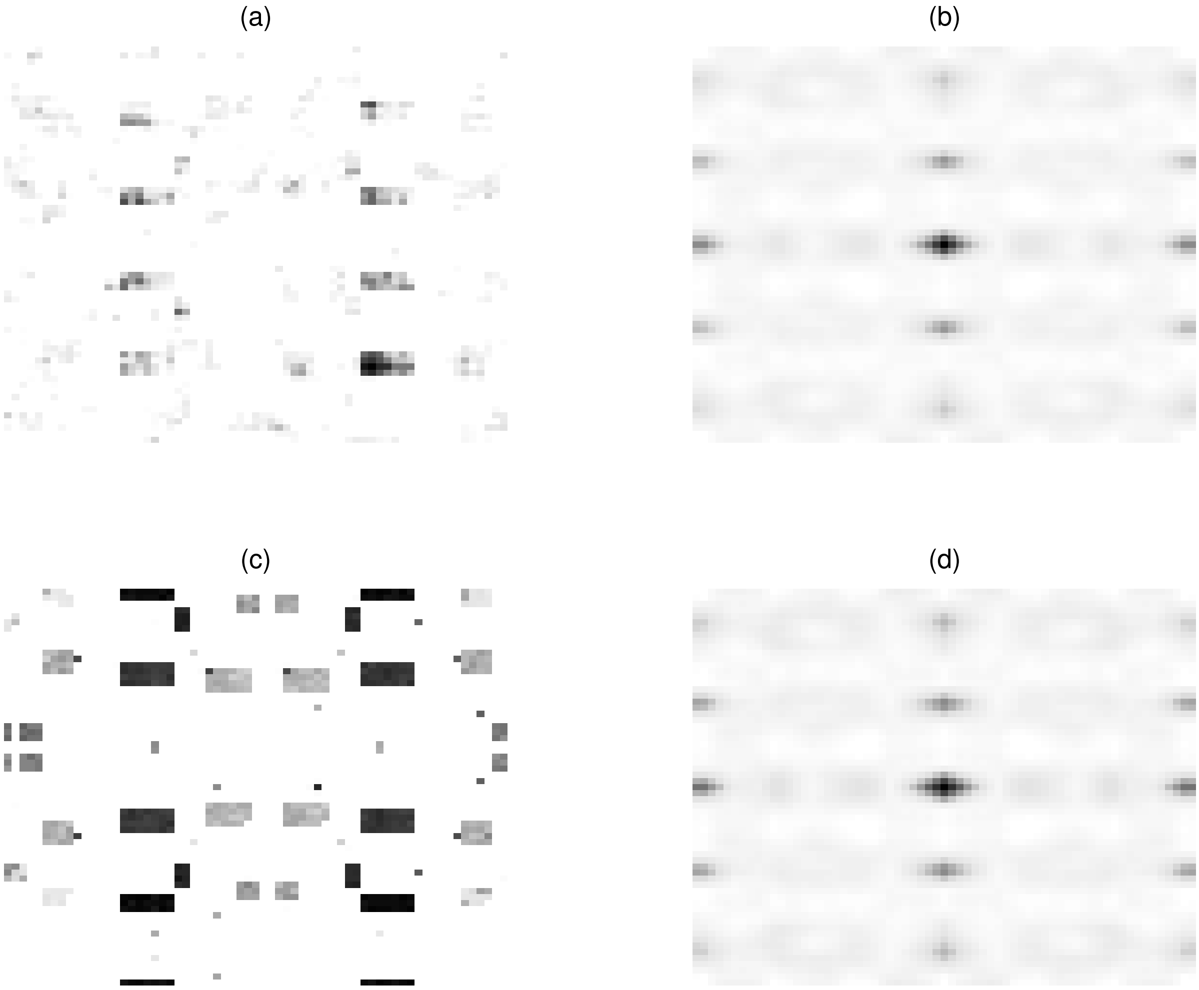


Figure 16: (a) Final estimate obtained when the initial estimate in Fig. 15(a) is used. (b) Patterson function of the initial estimate given in (a). (c) Final estimate when the initial estimate in Fig. 15(c) is used. (d) Patterson function of the initial estimate given in (c).

CHAPTER III

PRACTICAL CONCERNS ON THE APPLICATION OF MINIMUM I-DIVERGENCE METHODS TO X-RAY CRYSTALLOGRAPHY WITH REAL DATA

3.1 Introduction

We have proposed the application of minimum I-divergence methods to x-ray crystallography. As history tells us, x-ray crystallography is full of challenges, and finding a general, easy method for solving molecular structures remains a mystery.

The minimum I-divergence algorithms discussed in Chapter 2 suffer from two main challenges in their current forms. The most challenging problem is that the algorithms may converge to local minima that seem to have many different manifestations. The other main problem is the slow convergence speed of the algorithms. Both the original Schulz-Snyder iteration and our tweaked version for x-ray crystallography have multiplicative forms, which are notorious for their slow convergence. In x-ray crystallography, the problem is more serious because of the large amount of data in 3-D (compared with the 2-D case in astronomy.) Other concerns may also arise from these problems. For instance, finding good initial estimates is a difficult problem (although this is the case with all known practical phase retrieval algorithms.) Smarter choices for initial estimates would help alleviate the issue of local minima.

Despite all these current challenges, minimizing I-divergence seems to provide a good guiding principle for future research in x-ray crystallography. This chapter considers this aspect of minimum I-divergence methods from a practical point of view.

The Fourier magnitudes that crystallographers estimate never perfectly matches the measured Fourier magnitudes. This provokes curiosity about what electron density map the combination of the Fourier phases estimated by crystallographers and the measured Fourier magnitudes produce. We consider this question by manipulating the known information about protein 6PTI.

Crystallographers often appraise the quality of estimated molecular structures in terms of the so-called *R-factor* [108]. The R-factor is defined on the measured Fourier magnitudes and the estimated Fourier magnitudes, but the *I*-divergence is defined on their corresponding Patterson functions, which are related to the squared Fourier magnitudes. This complicated, indirect relation makes drawing an analytical conclusion difficult. For a particular numerical example involving the 6PTI protein, we will show that decreasing *I*-divergence corresponds with decreasing R-factor. For the best currently known estimated structure of the 6PTI protein, the R-factor is 0.1610.

Section 3.2 illustrates interesting crystallographic-data combinations of 6PTI and their associated electron density maps. The relationship between the R-factor and the *I*-divergence is presented in Section 3.3 from a practical perspective. We make some concluding remarks in Section 3.4.

3.2 Crystallographic Data of 6PTI

The 6PTI protein has space group $P21212$ and contains about 450 non-hydrogen atoms. The protein was measured at a resolution of 1.7 angstroms. We use this protein for the study in this chapter. The measured data and parameters estimated by crystallographers for this protein were obtained from the Protein Data Bank, managed by the Research Collaboratory for Structural Bioinformatics. [`http://www.rcsb.org`]

Figure 17 shows some selected slices from the measured Fourier magnitudes and the calculated Fourier magnitudes (computed by crystallographers) and their corresponding Patterson functions. The two Patterson functions look quite close to each other.

For the calculated Fourier magnitudes, the associated Fourier phases are known. Figure 18 shows some slices of ρ_{cal}, a $64 \times 64 \times 64$ electron density map of 6PTI. It is obtained by taking the inverse Fourier transform of the combination of the Fourier magnitudes and phases calculated by crystallographers:

$$\rho_{cal} = \mathscr{F}^{-1} \left\{ |F_{cal}| \exp \angle F_{cal} \right\}, \tag{29}$$

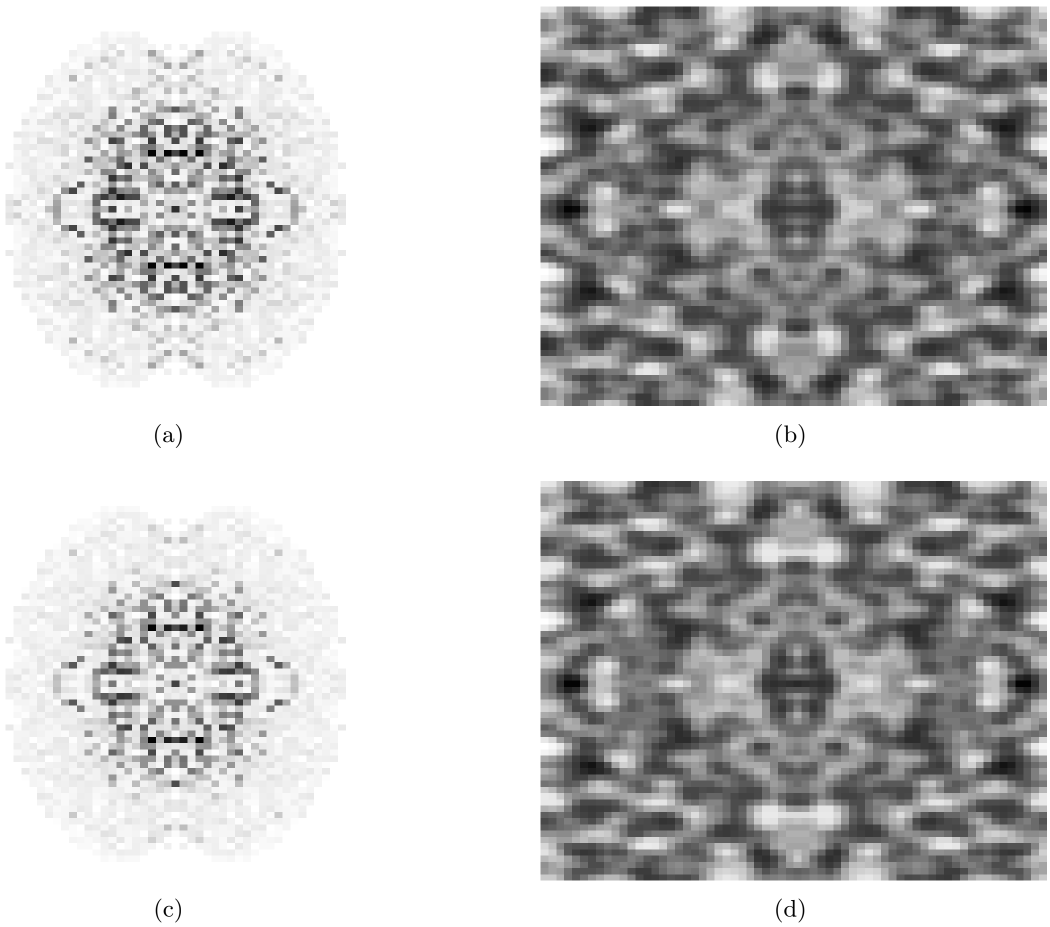
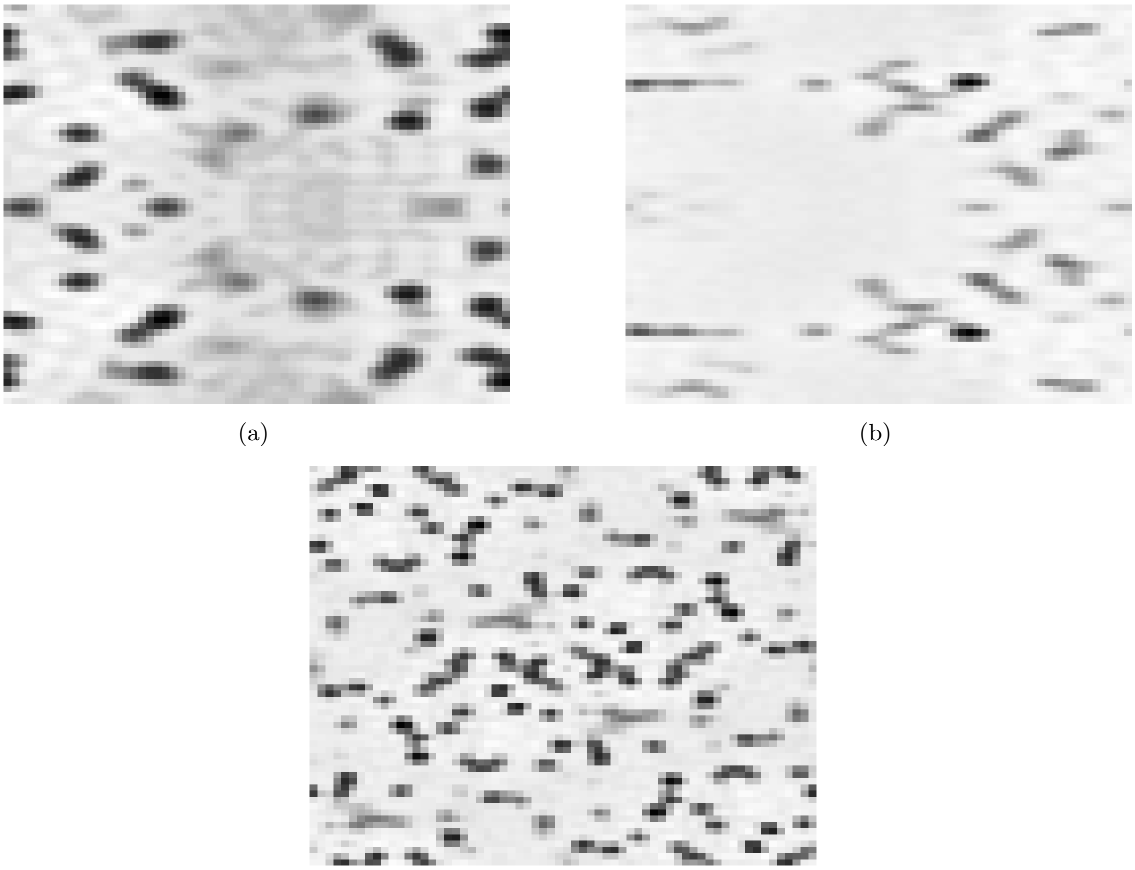


Figure 17: The data of protein 6PTI: (a) A slice of the measured Fourier magnitude (b) A slice of the Patterson function converted from the measured Fourier magnitude (c) A slice of the calculated Fourier magnitude by crystallographers (d) A slice of the Patterson converted from the calculated Fourier magnitude

where $|F_{cal}|$ and $\angle F_{cal}$ denote the calculated Fourier magnitudes and phases, respectively. Here, we should note that the calculated Fourier data are missing their DC components. Hence, the resulting electron density map sums to zero. For the reason, we add DC values that are large enough to raise all the components in an electron density map to nonnegative values.

There is some disparity between the calculated Fourier magnitudes and the measured Fourier magnitudes. It would be interesting to see how reasonable an image the *calculated* Fourier phases can produce when they are combined with the *measured* Fourier magnitudes, and how the electron density map associated with this combination differs from the ρ_{cal}.



(a) (b) (c)

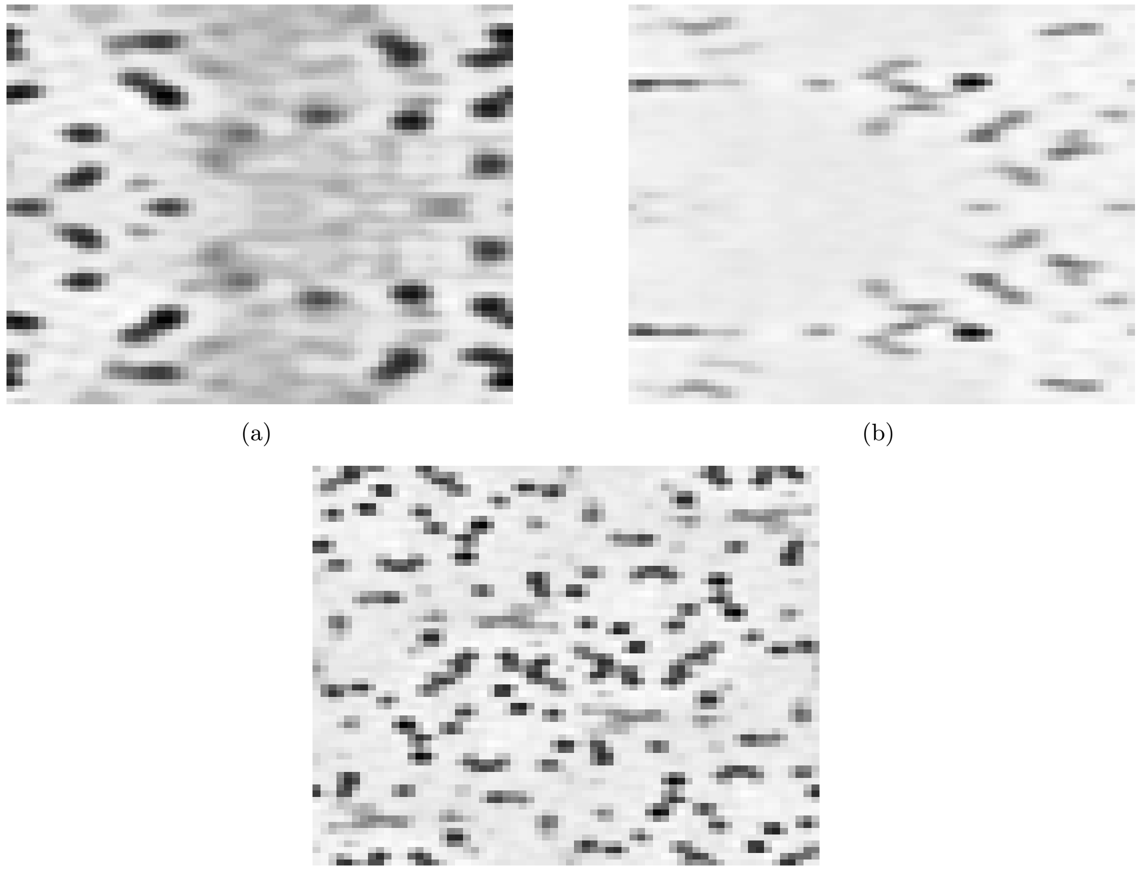
Figure 18: These figures show selected slices of ρ_{cal} of 6PTI taken along three different axes.

Let ρ_{syn} denote the electron density map created from the combination of the measured Fourier magnitudes and the calculated Fourier phases:

$$\rho_{syn} = \mathscr{F}^{-1} \left\{ |F_{obs}| \exp \angle F_{cal} \right\}, \tag{30}$$

where $|F_{obs}|$ denotes the measured Fourier magnitude. Figure 19 shows some slices of ρ_{syn}, which is a $64 \times 64 \times 64$ image. We can observe slight differences between the corresponding figures resulting from the disparity in the Fourier magnitudes.

Figure 20 presents images of the difference between the corresponding slices shown in Figs. 18 and 19. The maximum difference value between these two electron density maps is 0.0209, which is relatively large compared with the maximum values of ρ_{cal} (0.3283) and ρ_{syn} (0.3271); for reference, the associated minimum values are 0.0139 and 0.0134, respectively.



(a) (b) (c)

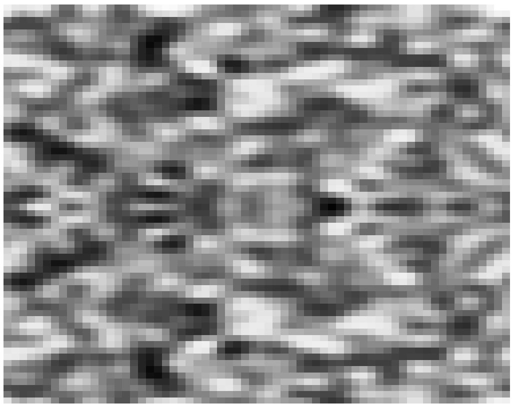
Figure 19: These figures show selected slices of ρ_{syn} of 6PTI taken along three different axes. These slices correspond to the slices in Figure 18.

3.3 Discussion on the R-factor and the I-divergence

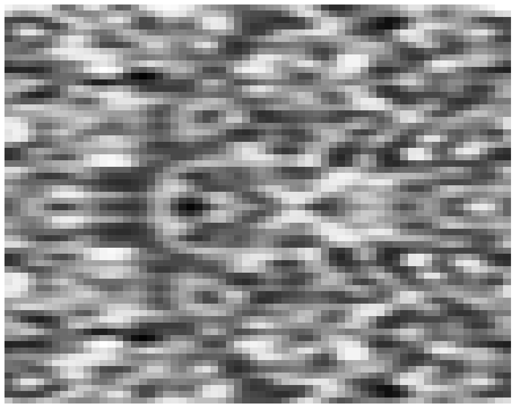
When crystallographers come up with a candidate molecular structure, they often assess the quality of the associated electron density map by the R-factor. The R-factor measures agreement between the measured Fourier magnitudes and the calculated Fourier magnitudes. When this R-factor is less than 0.5, the estimated molecular structure is considered to be reasonably close to the true structure. This reasonable structure is further refined with various refinement methods to obtain a better R-factor value. The R-factor is given by

$$R(F_{obs}, F_{cal}) = \frac{\sum_{\mathbf{h}} \big| |F_{obs}(\mathbf{h})| - |F_{cal}(\mathbf{h})| \big|}{\sum_{\mathbf{h}} |F_{obs}(\mathbf{h})|}. \tag{31}$$

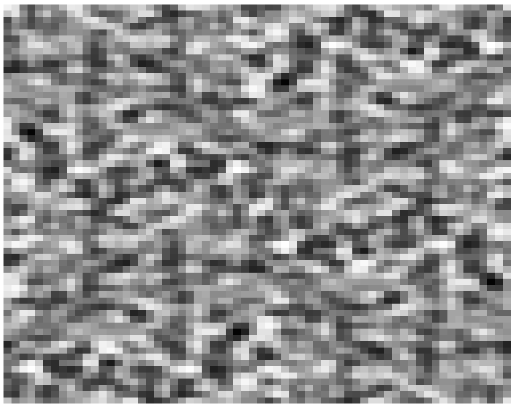
For the 6PTI protein, the best known R-factor value in literature is 0.1610, as indicated



(a)



(b)



(c)

Figure 20: These figures show the difference between the corresponding panels in Figures 18 and 19.

by the data profile provided on the Protein Data Bank. We show that our minimum *I*-divergence algorithm can further improve this R-factor value.

We initialized the algorithm with the ρ_{cal} of 6PTI, shown in Fig. 18. Here, we are investigating the algorithm as a way of refining an existing guess of the electron density map. This also lets us focus on R-factor issues independent of questions about local minima. Figure 21 shows graphs that illustrate changes in the R-factor and the *I*-divergence values as iteration proceeds. As shown in the graphs, both of the functions are monotonically decreasing. Table 5 shows the R-factor and the *I*-divergence of some selected iterations.

Let ρ_{est} denote the estimate of electron density map produced by our minimum *I*-divergence algorithm. For comparison, Figs. 22, 23, and 24 show some slices of ρ_{syn} and images of differences between the slices of ρ_{syn} and the corresponding slices of ρ_{est} and ρ_{cal}

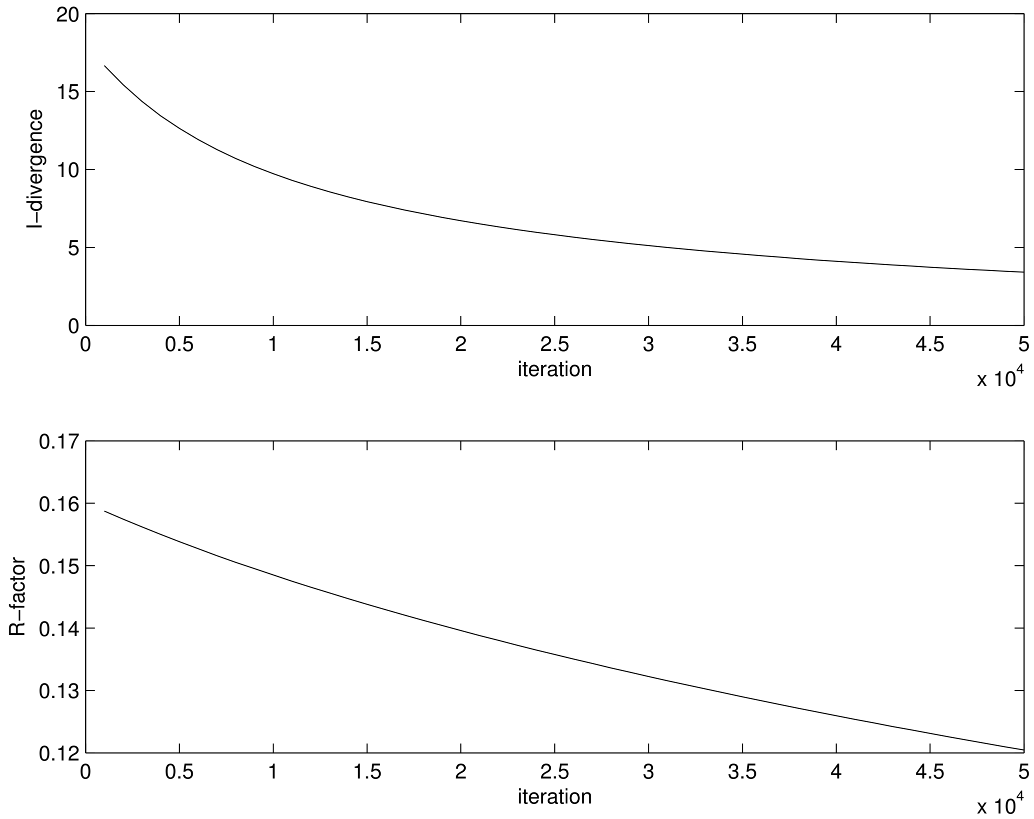


Figure 21: Comparison of the changes of the R-factor and the *I*-divergence when our minimum *I*-divergence algorithms is initialized with the ρ_{cal} of 6PTI.

taken along three different axes, respectively. The 50000-*th* iteration estimate was chosen for the ρ_{est}. Recall that ρ_{syn} is obtained by using the measured Fourier magnitude. While they all look very close to each other visually, it is intriguing that ρ_{est} is closer to ρ_{cal} than ρ_{syn}: The sum of the squared difference between ρ_{est} and ρ_{cal} is 1.0849 compared with 3.2347 between ρ_{est} and ρ_{syn}, and the sum of the absolute value of the difference between ρ_{est} and ρ_{cal} is 405.9355 compared with 731.6872 between ρ_{est} and ρ_{syn}.

3.4 Conclusions

Typically, there is some mismatch between the crystallographer's calculated Fourier data and the measured Fourier data. We have explored what kind of mismatch may exist between these data by combining the measured Fourier magnitudes with the calculated Fourier phases using an example involving protein 6PTI.

In crystallographic literature, such mismatch is described and measured by the R-factor.

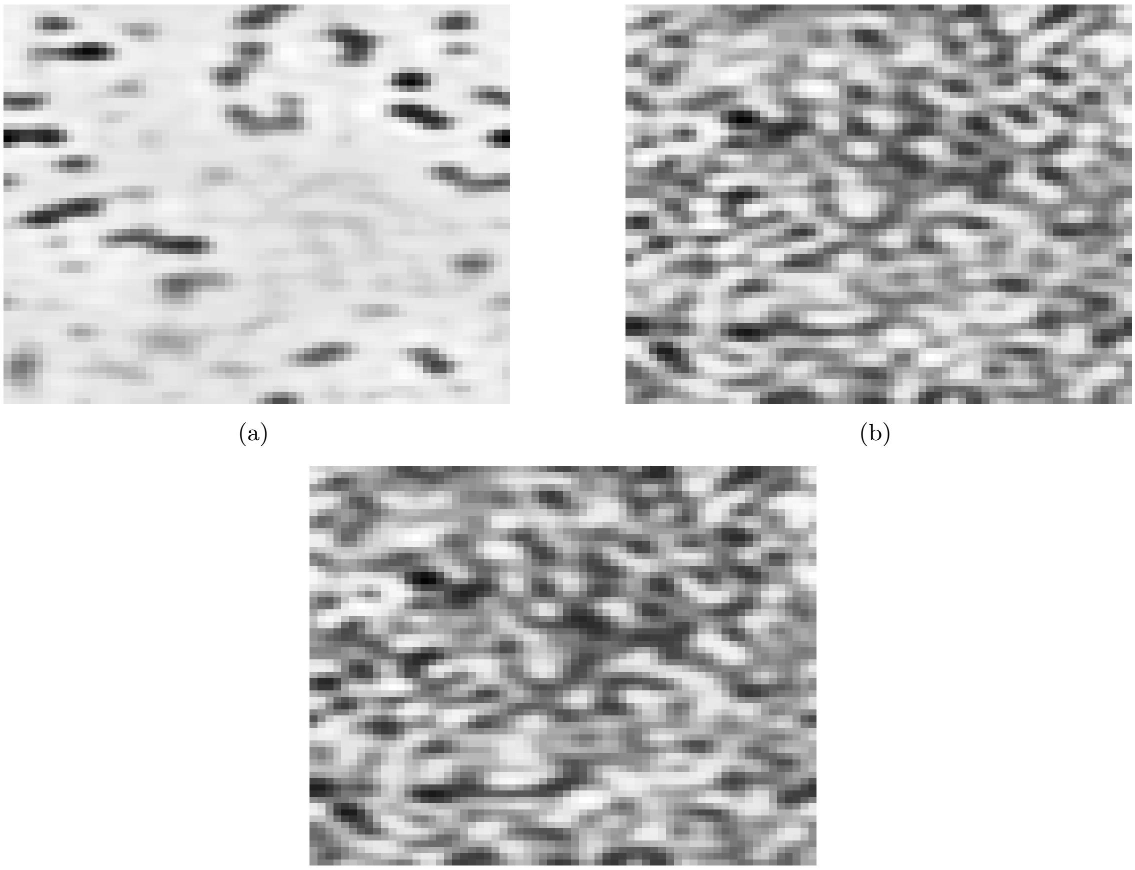
Table 5: Comparison of R-factor and I-divergence.

Iteration	R-factor	I-divergence
1000	0.1588	16.6672
2000	0.1575	15.4369
3000	0.1562	14.3732
4000	0.1550	13.4468
5000	0.1538	12.6342
6000	0.1527	11.9167
7000	0.1516	11.2789
8000	0.1505	10.7084
9000	0.1495	10.1953
10000	0.1485	9.7311
15000	0.1438	7.9416
20000	0.1396	6.7153
25000	0.1358	5.8142
30000	0.1322	5.1200
35000	0.1290	4.5665
40000	0.1260	4.1134
45000	0.1231	3.7348
50000	0.1205	3.4131

For a particular example, we showed that when the I-divergence decreased, the R-factor consistently decreased as well. This is a promising result in the sense that the minimizing I-divergence, which is not well-known in the crystallographic literature, corresponded with minimizing a measure crystallographers are used to dealing with. We emphasize that in our numerical example, we further refined the known electron density map with an R-factor of 0.1610 to a map with an R-factor of 0.1205.[1]

If a minimum I-divergence algorithm that can consistently converge to a global minimum can be found, then minimizing I-divergence would ultimately lead to the correct electron density map. Therefore, useful agenda for future research would be to find methods to avoid local minima.

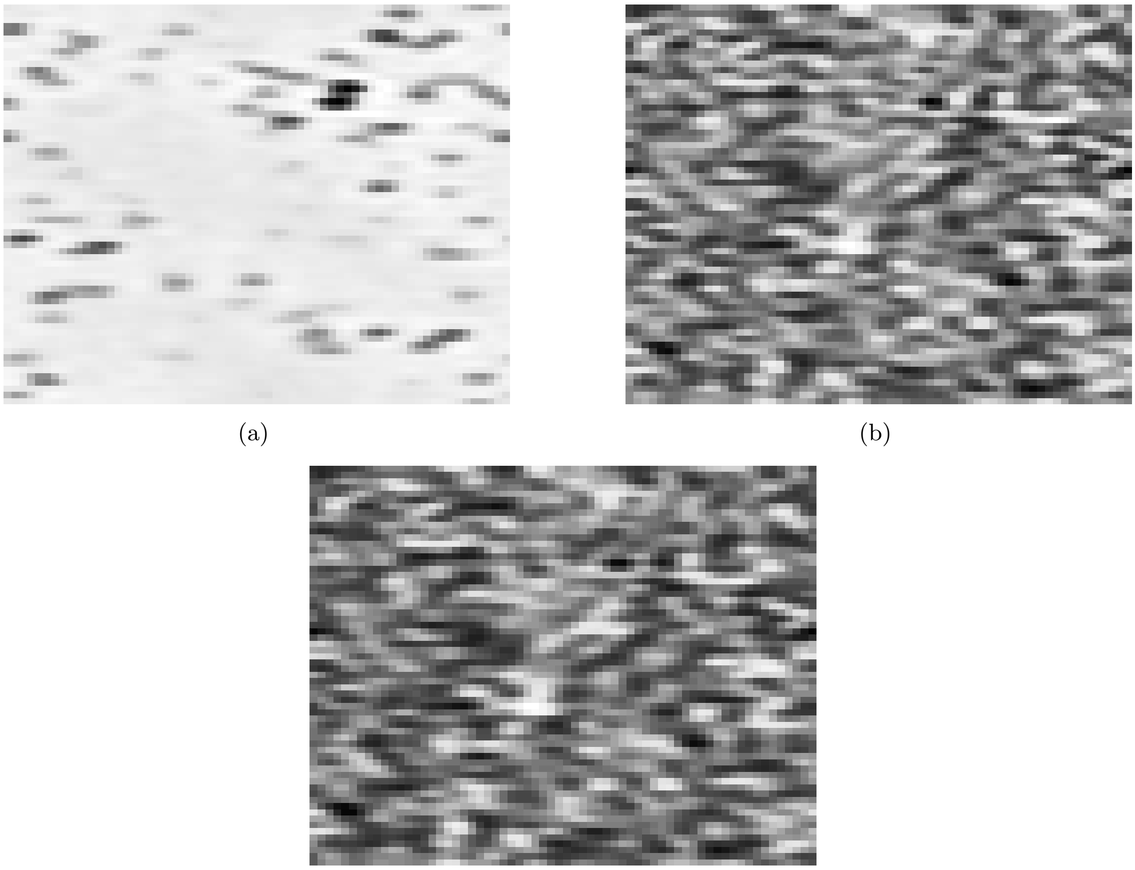
[1]The R-factor is not the only criterion by which crystallographers evaluate a candidate structure. Molecules are subject to various forces that constrain the relations betweens atoms. Crystallographers often follow a multi-stage procedure wherein they: 1) develop a rough electron density map, 2) manually fit molecular models to that map, and 3) refine their molecular models further using the original diffraction data combined with detailed models of atomic interactions derived from physics. In this broader context, our algorithms may be thought of as part of step (1), namely obtaining an initial electron density map. The remaining refinements involve chemical models – for instance, specific amino acid models.



(a) (b) (c)

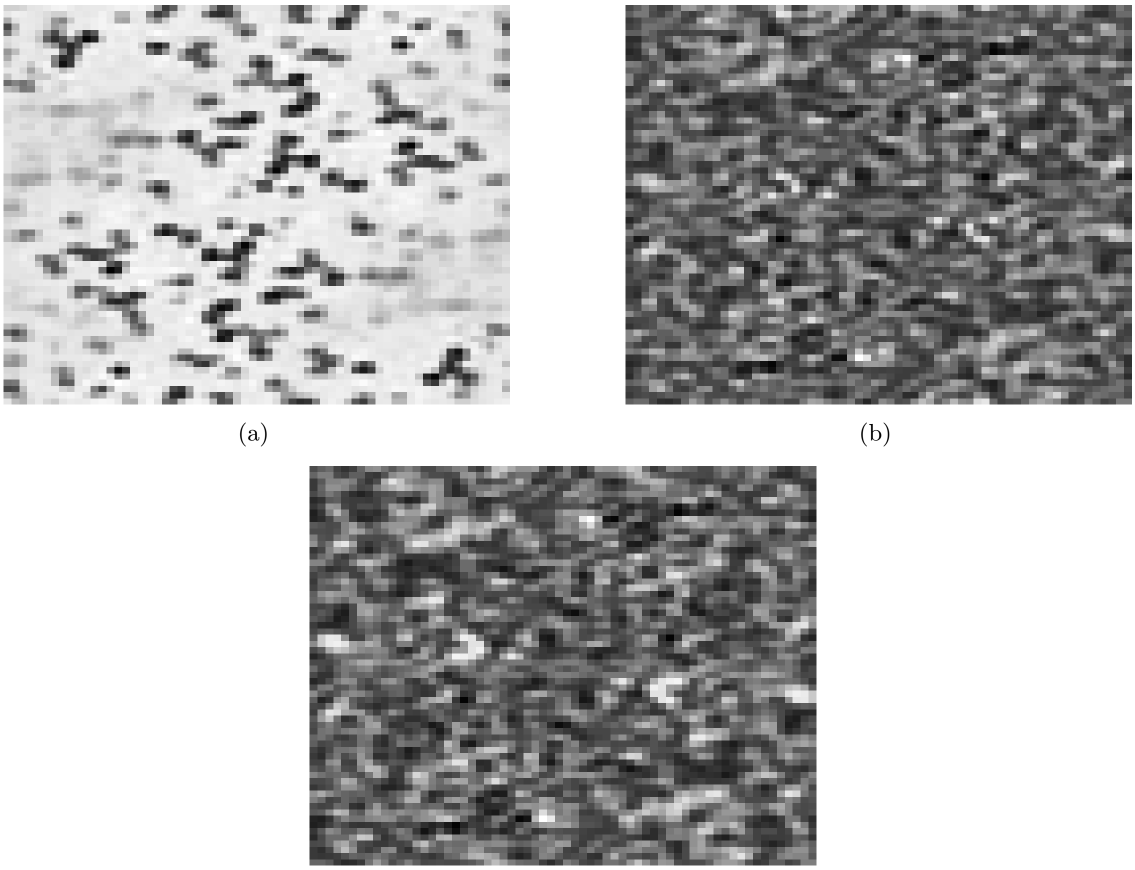
Figure 22: (a) A slice of the ρ_{syn} of 6PTI. (b) Image of the difference between the slice in (a) and the same slice of the ρ_{est} of 6PTI at the 50000-*th* iteration. (c) Image of the difference between the slice in (a) and the same slice of the ρ_{cal} of 6PTI. (Note: Because the slices of ρ_{syn}, ρ_{est}, and ρ_{cal} are visually identical, we show the slice of ρ_{syn} and differences between the slice and the corresponding slices of ρ_{est} and ρ_{cal}.)

Another issue is the slow convergence of our algorithms in their current forms, as shown in our numerical example. This makes it hard to interact and experiment with the algorithms. Hence, another useful avenue for future work would be to find algorithms with faster convergence. The minimum *I*-divergence algorithms originated from the corresponding EM algorithms involving a specific type of incomplete data. That is, the minimum *I*-divergence algorithms are deterministic forms of their corresponding EM algorithms. Based on this aspect, we may be able to employ some acceleration techniques used for EM algorithms such as SAGE [31], PX-EM [71], the methods by Jamshidian and Jennrich [49, 50], and ECME [70].



(a) (b) (c)

Figure 23: (a) A slice of the ρ_{syn} of 6PTI. (b) Image of the difference between the slice in (a) and the same slice of the ρ_{est} of 6PTI at the 50000-*th* iteration. (c) Image of the difference between the slice in (a) and the same slice of the ρ_{cal} of 6PTI. (Note: Because the slices of ρ_{syn}, ρ_{est}, and ρ_{cal} are visually identical, we show the slice of ρ_{syn} and differences between the slice and the corresponding slices of ρ_{est} and ρ_{cal}.)



(a) (b) (c)

Figure 24: (a) A slice of the ρ_{syn} of 6PTI. (b) Image of the difference between the slice in (a) and the same slice of the ρ_{est} of 6PTI at the 50000-*th* iteration. (c) Image of the difference between the slice in (a) and the same slice of the ρ_{cal} of 6PTI. (Note: Because the slices of ρ_{syn}, ρ_{est}, and ρ_{cal} are visually identical, we show the slice of ρ_{syn} and differences between the slice and the corresponding slices of ρ_{est} and ρ_{cal}.)

CHAPTER IV

ON CONVERGENCE TO LOCAL MINIMA OF THE SCHULZ-SNYDER PHASE RETRIEVAL ALGORITHM

4.1 Introduction

The goal of phase retrieval is to infer Fourier phase information given only Fourier magnitude data [32, 93]. Phase retrieval is quite challenging. Phase retrieval is equivalent to the problem of recovering a function from its autocorrelation. One fundamental problem is that there may be multiple signals whose autocorrelations are the same. Moreover, for iterative algorithms intended to minimize an objective function, the algorithm may converge to local minima that do not correspond with a global minimum. In addition to these problems, some iterative phase retrieval algorithms may suffer from stagnation, where the algorithm gets stuck on an estimate that does not correspond to a local minimum of the objective function [34]. The stagnation problems in Fienup's algorithm, which seems to be the most widely known of the phase retrieval methods, have been discussed by Fienup and Wackerman [34]. This chapter investigates the problem of convergence to local minima for a particular technique, which we call the Schulz-Snyder algorithm.

Schulz and Snyder [93] developed an iterative method for phase retrieval that works entirely in the spatial domain, instead of alternating between the frequency and spatial domains as in Fienup's algorithm. Although Schulz and Snyder did not claim that their algorithm would never be subject to convergence to local minima (as opposed to a global minimum), they were encouraged by the fact that no such problems were observed in any of their experiments. Unfortunately, we have found that for certain cases, *the Schulz-Snyder algorithm can converge to an incorrect solution.* The Schulz-Snyder algorithm is based on minimizing an information-theoretic distance. We illustrate that incorrect solutions correspond to local minima on the surface of this objective function. This offers the hope of improving the algorithm by providing ways for it to escape local minima. To our knowledge, this chapter is the first to report on and characterize the problem of convergence to local minima for this algorithm.

Since an analytic proof ensuring convergence of the iteration to a local minimum (as opposed to a saddle point) does not guarantee that a practical solution, obtained by the algorithm in a finite number of iterations, is a local minimum for sure, we take a numerical approach to confirming convergence to local minima. Our approach checks a set of sufficient conditions for a local minimum that involve the first and second derivatives of the objective function being minimized. All of our experiments use simulated data, so we can plug in the "truth" as the initial estimate to find a true global minimum, and then explore additional local minima.

This chapter is organized as follows. Section 4.2 reviews the Schulz-Snyder algorithm and defines notation that will be used throughout this chapter. Sufficient conditions for local minima of the objective function surface are presented in Section 4.3, while numerical experiments illustrating local minima are given in Section 4.4. Our discussion is then concluded in Section 4.5.

4.2 The Schulz-Snyder Algorithm

The Schulz-Snyder algorithm [93] is an iterative method for recovering nonnegative functions from their n-th order correlations. Here, we focus specifically on the $n = 2$ case of recovery from autocorrelations, which is equivalent to phase retrieval. The algorithm estimates images from their autocorrelations by minimizing an information-theoretic distance between measured autocorrelation data and the autocorrelation of the estimated image. The distance used is Csiszar's I-divergence [23], which is a generalization of the Kullback-Leibler distance [62]. It is defined as

$$D[S, R_f] = \sum_y [R_f(y) - S(y)] + \sum_y S(y) \ln \frac{S(y)}{R_f(y)}, \tag{32}$$

where $S = R_g$ is the autocorrelation of some true but unknown g that we want to reconstruct from S, and the autocorrelation of an estimate f is defined as

$$R_f(y) = \sum_x f(x)f(x+y). \tag{33}$$

The purpose of the algorithm is to minimize the following objective function:

$$\begin{aligned} J(f) &= D[S, R_f] \\ &= \sum_y [R_f(y) - S(y)] + \sum_y S(y) \ln \frac{S(y)}{R_f(y)}, \end{aligned} \tag{34}$$

subject to the constraints

$$\begin{aligned} I(f) = \sum_y f(y) &= I(g) \\ f \geq 0, & \end{aligned} \tag{35}$$

where $I(f)^2 = \sum_y R_f(y)$, which results from Property 3.4 in Schulz and Snyder [93]. Note that $I(g)$ can be obtained even if g is not known.

The Schulz-Snyder algorithm for recovering a nonnegative function from its autocorrelation is specified by the iteration

$$\begin{aligned} f_{k+1}(x) &= f_k(x) \frac{1}{2I(f)} \sum [f_k(x+y) + f_k(x-y)] \frac{S(y)}{R_{f_k}(y)} \\ &= f_k(x) \frac{1}{I(f)} \sum f_k(x+y) \frac{[S(y) + S(-y)]}{2R_{f_k}(y)}. \end{aligned} \tag{36}$$

Note that if $f_0(x) = 0$ for some particular x, then $f_k(x) = 0$ for that x for all k. This provides a convenient way of incorporating support constraints if they are available.

4.3 Sufficient Conditions for Local Minima

This section develops criteria for determining whether or not the algorithm has indeed converged to a local minimum. In practice, it is not automatically guaranteed that the final estimate is a local minimum, even when the convergence of the algorithm, given a theoretically unlimited number of iterations, is assured by an analytic proof. This may be because the final estimate is a saddle point, or the algorithm may improve the estimates too slowly due to finite-precision numerical issues or the small curvature at the local minimum toward which the algorithm is headed. For these reasons, we suggest a set of sufficient conditions that can establish local minimality of a given estimate.

Generally speaking, it would be extremely difficult to tell whether or not a local minimum is a global minimum. For this reason, we conduct experiments where the true images are known, so that we may know for sure that plugging the "right answer" into the algorithm corresponds to a global minimum.

Before presenting the sufficient conditions, we define two sets that will be useful in later discussions.

Definition 4. *The two sets $\mathcal{S}_1$ and $\mathcal{S}_2$ are called* index sets *and defined as follows:*

$$\mathcal{S}_1 = \left\{ i \ : \ \frac{\partial J(f)}{\partial f(x_i)} \ = \ 0, \ \ x_i \in \chi \right\}, \tag{37}$$

$$\mathcal{S}_2 = \left\{ i \ : \ \frac{\partial J(f)}{\partial f(x_i)} \ > \ 0, \ \ x_i \in \chi \right\}, \tag{38}$$

where χ represents the two-dimensional set $\{1, 2, \ldots, N\} \times \{1, 2, \ldots, M\}$.

The I-divergence in the Schulz-Snyder algorithm is guaranteed to be non-increasing as iterations proceed [93]. Hence, if the algorithm has converged to a critical point, it cannot be a maximum; it must be either a minimum or a saddle point. Therefore, at convergence, $\mathcal{S}_1 \bigcup \mathcal{S}_2 = \mathcal{S}$, where $\mathcal{S}$ is the set of all the parameter indices.

The sufficient conditions for local minimality of a given estimate are given in the following theorem.

Theorem 3. (Sufficient conditions for a local minimum) *If an estimate f^* satisfies the following conditions:*

1. *The given estimate satisfies the Kuhn-Tucker conditions:*

$$\left. \frac{\partial J(f)}{\partial f(x)} \right|_{f=f^*} \begin{cases} = 0 \ f^*(x) > 0 \\ \geq 0 \ f^*(x) = 0. \end{cases} \tag{39}$$

2. *The Hessian matrix H whose (i,j)-th element is defined by*

$$H_{ij} = \left. \frac{\partial^2 J(f)}{\partial f(x_i) \partial f(x_j)} \right|_{f=f^*}, \ \ i \in \mathcal{S}_1, \ \ j \in \mathcal{S}_1 \tag{40}$$

   is positive definite,

*then f^* is a local minimum. In other words, for any f that satisfies the given constraints (*i.e., *nonnegativity) and is in the neighborhood of f^*, the following inequality is satisfied:*

$$D[S, R_{f^*+\Delta f}] - D[S, R_{f^*}] = J(f^* + \Delta f) - J(f^*) > 0, \tag{41}$$

where $\Delta f = f - f^*$.

Proof. See Appendix B.1. □

Although the conditions just mentioned are clear theoretically, they may involve various numerical issues in practice, such as determining when quantities are numerically zero (which influences construction of the two index sets) and the precision of the algorithm. Because such issues are quite problem-dependent and cannot be resolved by a single general rule, we address the issues based on problems under consideration in a reasonable way that takes experiments into consideration.

The first derivative of the I-divergence, which is implicitly embedded in the algorithm, has been derived by Schulz and Snyder:

$$\frac{\partial J(f)}{\partial f(x_i)} = 2I(f) - \sum_y \{f(x_i + y) + f(x_i - y)\} \frac{S(y)}{R_f(y)}. \tag{42}$$

In this equation, the x_i can be in either $\mathcal{S}_1$ or $\mathcal{S}_2$. The second partial derivative, derived in Appendix B.2, is given by

$$\begin{aligned} \frac{\partial^2 J(f)}{\partial f(x_i) \partial f(x_j)} &= 2 + \sum_y \{f(x_i + y) + f(x_i - y)\}\{f(x_j + y) + f(x_j - y)\} \frac{S(y)}{R_f(y)} \\ &\quad - \left(\frac{2S(x_j - x_i)}{R_f(x_j - x_i)} \right). \end{aligned} \tag{43}$$

4.4 Experiments

4.4.1 Preliminary Remarks

4.4.1.1 Convergence Criteria

Due to the buildup of numerical errors, an iterative algorithm implemented on a computer with finite-precision arithmetic may not *exactly* settle on a fixed-point solution. The algorithm may wander slightly about a point at which it is supposed to stabilize, or take excruciatingly tiny steps towards the point. For these reasons, it is important to establish a meaningful stopping criterion before discussing other matters. Various choices appear in literature. We are tempted to choose I-divergence values in our stopping criterion. However, we have observed that the algorithm may make further improvements in solutions even while the numerically computed I-divergence wanders up and down slightly about a value. Hence, we choose a stopping criterion based on the pixel values themselves:

Stoppping Criterion 1. *The algorithm is halted when the following condition is satisfied:*

$$\max_i |f_k(x_i) - f_{k-1}(x_i)| < \epsilon, \tag{44}$$

where ϵ is a positive constant.

The choice of epsilon depends on the experiments to be performed. It may be related to precisions of the operations involved in the algorithm. Although the epsilon may be selected by analyzing the possible errors (or precisions), it would be quite difficult to estimate all the errors in advance. Since we know the "truth" in our experiments, we had the benefit of choosing appropriate epsilons by plugging the truth into the algorithm and seeing how far the estimate slides off of the truth due to numerical issues. This chapter focuses on probing the limits of the algorithm, and hence we select our epsilons rather strictly. In practice, we would recommend epsilons that are are larger, i.e. looser, than those used in this study.

Table 6 shows the epsilon values used for the experiments in this study. In the following tables, the results of one experiment include the original image and estimates, autocorrelations, gradient images, and line plots (if contained). For instance, the results of Exp. 1, abbreviating Experiment 1, are shown in the set of images in Fig. 1 through Fig. 3, and the results of Exp. 7 are shown in the set of images in Fig. 22 through Fig. 25.

Table 6: Choices of epsilon for the experiments in this study

	Exp. 1	Exp. 2	Exp. 3	Exp. 4	Exp. 5	Exp. 6	Exp. 7	Exp. 8
ϵ	10^{-13}	10^{-13}	10^{-7}	10^{-7}	10^{-7}	10^{-8}	10^{-7}	10^{-8}

Note that the chosen epsilons are not the same for all the experiments. For instance, in Exps. 1 and 2, the algorithm sometimes require far more iterations (compare Tables 10 and

11 with Tables 12 to 16) than the others, and we need a rather precise epsilon to convince ourselves that the algorithm has surely converged. Note that it is not necessarily true that we need smaller epsilons for larger parameter spaces; compare Exps. 1 and 2 with Exp. 3 (see Table 6). It is also not true that smaller epsilons always require the algorithm to take more iterations, as seen by comparing Exp. 3 with Exp. 6 (see Tables 12 and 15). The choice seems to depend on where a local minimum is located on the surface of the I-divergence function, which we cannot know until the algorithm converges. As just mentioned, it would be extremely difficult to predict what choices of the epsilon are appropriate ahead of time, since the choices may depend on different aspects of the algorithm in unfathomable ways. Therefore, we determine them by trial and error.

Numerical concerns may be sensitive to implementation issues. All our experiments were performed with MATLAB 6.5 from The MathWorks on a machine with a 2.0 GHz Intel Pentium IV. The precision of the machine is 10^{-16}. This number is determined by selecting the maximum δ such that $1 + \delta = 1$ in a given machine. Any δ smaller than 10^{-16} added to 1 produces 1, and any δ larger than 10^{-16} added to 1 produces a larger number than 1.

4.4.1.2 Initial Estimates

Throughout the chapter, *unless stated otherwise*, the algorithm is initialized with a rectangle with constant intensity plus a small amount of uniform random noise (we use "constant + 0.1×random noise"), which is scaled according to the constraint in Eq. (35). In our experiments, the selected constant is 1, and the uniform random noise takes values on between 0 and 1. This small amount of noise is helpful to prevent the algorithm from getting stuck on artificially symmetric images, as discussed in the right column of [93, p. 1269]. One might wonder if different realizations of this small additive noise would result in the algorithm converging to radically different answers. In all our experiments, we never found this to be the case. We found that significantly different initial estimates were needed to obtain different results.

The sizes of the initial estimates are chosen to be as small as possible to minimize accumulated computational errors resulting from a large number of parameters. The size of the initial images for all experiments except for Exp. 3 is chosen to be 16 $\times$ 16 pixels because images of the size manifest the effects that we want to show (i.e., local minima). On the other hand, the size for Exp. 3 is chosen to be 32 $\times$ 32 pixels because initial images smaller than that did not produce the effects that we want to illustrate.

When the initial estimates are put into the algorithm, they are zero-padded for ease of implementation; we implement the correlation operations in the Schulz-Snyder algorithm with fast Fourier transforms. The sizes of all the original true images are the same as those of the corresponding initial images. In the images shown throughout the chapter, we show only a subset of the full image to illustrate detail; surrounding pixels may be considered to be zero. Brighter pixels represent smaller values, and darker pixels represent larger values. Also, all the images in this chapter are scaled to fit the full dynamic range when they are displayed.

We show the gradient image of the initial estimate for each experiment in Figs. 3(b), 6(b), 9(b), 12(b), 16(b), 20(b), 24(b) and 28(b). The gradient images are also scaled to use the full dynamic range, hence it may be difficult to compare pixel values from image to image. Hence, Table 7 shows the range of the gradient values. Furthermore, we add a column containing the number of gradient values between -0.1 and 0.1. The ranges and numbers may vary slightly with the choice of initial images, but not significantly. In Table 7, f_0 denotes an initial estimate, and ∇ denotes gradient. The elements of the set $\mathcal{A}$ are the gradient values between -0.1 and 0.1, and the symbol $|\mathcal{A}|$ denotes the cardinality of $\mathcal{A}$.

Functions shifted and/or rotated by 180 degrees have exactly the same autocorrelations as the original function. Therefore, the final estimates of some experiments are shifted and/or rotated by 180 degrees from their original versions (*e.g.*, see Figs. 7 and 10).

4.4.1.3 Numerical Zero

Another issue we need to be careful about is the construction of the index sets $\mathcal{S}_1$ and $\mathcal{S}_2$. It may be controversial to say whether a gradient should be considered numerically zero or not. Hence, we take into account not only the gradient values but also the values of the

Table 7: The range of gradient values for the initial images, and the number of the gradient values between -0.1 and 0.1.

	$\max \nabla f_0$	$\min \nabla f_0$	$\|\mathcal{A}\|$
Exp. 1	58.66	-39.81	1
Exp. 2	86.33	-17.62	2
Exp. 3	305.99	254.64	0
Exp. 4	27.76	-27.14	1
Exp. 5	35.51	-32.58	2
Exp. 6	36.55	-28.74	1
Exp. 7	31.52	-20.75	2
Exp. 8	34.08	-20.79	2

final estimates. Recall that if the algorithm has converged, then nonzero-valued pixels of an estimate are supposed to have zero-gradient values. Therefore, we construct $\mathcal{S}_1$ with the "nonzero" pixels. Determination of the nonzeroness of pixels may also be controversial. We conservatively consider pixels whose values are greater than 10^{-4} to be the "nonzero" pixels. Now, the gradient values corresponding to the nonzero pixels should be zero in theory. However, numerical limitations keep the values from becoming exactly zero. Table 8 shows the maximum and minimum of the gradient values computed on the index set $\mathcal{S}_1$ in each experiment. As seen on the table, the gradient values are close to zero; if we run the algorithm further, then the values become smaller.

Table 8: Maximum and minimum of the gradient values for the final estimate corresponding to the index set $\mathcal{S}_1$ in each experiment.

	$\max \nabla f(\mathcal{S}_1)$	$\min \nabla f(\mathcal{S}_1)$
Exp. 1	1.2037×10^{-12}	5.1563×10^{-12}
Exp. 2	4.0412×10^{-13}	9.3425×10^{-13}
Exp. 3	9.3342×10^{-8}	9.7413×10^{-8}
Exp. 4	1.6614×10^{-7}	8.0929×10^{-7}
Exp. 5	2.0553×10^{-7}	3.4158×10^{-7}
Exp. 6	3.8334×10^{-8}	2.8034×10^{-6}
Exp. 7	8.8482×10^{-8}	1.0549×10^{-7}
Exp. 8	3.2379×10^{-6}	3.5768×10^{-9}

To convince ourselves that the construction of the index set $\mathcal{S}_1$ is correct, we investigate the gradient values computed on the zero-value index set $\mathcal{S}_2$ to see if there are any values with gradient values in the ranges given in Table 8. Table 9 shows the maximums of the estimated values corresponding to the index set $\mathcal{S}_2$, along with the associated gradient values. The estimated values corresponding to the index set $\mathcal{S}_2$ are sufficiently small to claim their zeroness, and the associated gradient values are large enough to claim their nonzeroness. With low probability, there might be some pixels whose values are close to zero with gradient values close to zero, according to the theory of calculus. However, we have not met such pixels in any of our experiments. None of the gradient values associated with the index set $\mathcal{S}_2$ fell in the ranges given in Table 8.

Table 9: Maximum of the values of the final estimate corresponding to the index set $\mathcal{S}_2$, along with the associated gradient value in each experiment.

	$\max f(\mathcal{S}_2)$	$\nabla\{\max f(\mathcal{S}_2)\}$
Exp. 1	1.7509×10^{-9}	5.2072×10^{-5}
Exp. 2	6.3692×10^{-12}	1.8345×10^{-4}
Exp. 3	4.5767×10^{-149}	8.8046×10^{-3}
Exp. 4	4.7416×10^{-166}	6.5779×10^{-3}
Exp. 5	6.0390×10^{-167}	2.5424×10^{-2}
Exp. 6	9.0652×10^{-9}	1.2225×10^{-4}
Exp. 7	4.4466×10^{-323}	5.2912×10^{-2}
Exp. 8	1.1884×10^{-7}	3.3057×10^{-3}

4.4.1.4 Positive Definiteness

Positive definiteness of the Hessian matrices is tested by checking the eigenvalues of the matrices. The eigenvalues are computed with the MATLAB command "eig." Even though we computed all the eigenvalues associated with each matrix, we only show the minimum and maximum eigenvalues for brevity.

4.4.2 Radically Different Results

4.4.2.1 Experiment 1

Figure 25(a) shows an original image, which is quite simple, and Fig. 25(c) shows the final estimate of the original image reconstructed by the Schulz-Snyder algorithm. As mentioned, the final estimate is obtained by halting the algorithm when the given stopping criterion is met. The estimate looks fairly different from the original image. Because the estimate stayed visually changeless even after many more iterations, we suspected that the estimate was one of the local minima. To confirm this conjecture, we test the sufficient conditions discussed in the preceding section. Table 10 shows I-divergence values for the original true image, the initial estimate, and the final estimate, denoted by $D[S, S]$, $D[S, R_{f0}]$, and $D[S, R_{f_k}]$, respectively, where f_k denotes the final estimate, and k is the number of iterations when the final estimate is obtained. The I-divergence value for the original true function is supposed to be zero. However, due to numerical errors, it moved about the I-divergence value given in Table 6 as iterations proceeded; the value is recorded after one iteration. Note that the I-divergence value for the final estimate is relatively large in comparison with the I-divergence value for the truth, even though the image size is small. The value of the norm used in the stopping criterion, $\max_i |f_k(x_i) - f_{k-1}(x_i)|$, is given. The information on the Hessian matrix described in the sufficient conditions is also contained. Based on the minimum eigenvalue of the matrix, the Hessian matrix formed from the set $\mathcal{S}_1$ is positive definite. The size of the Hessian matrix is also given with other information.

Table 10: Selected data from Exp. 1.

Quantity	Value
$D[S, S]$	6.0646×10^{-13}
$D[S, R_{f_0}]$	3220.7138
$D[S, R_{f_k}]$	4.5947
k	308320
$\max_i \|f_k(x_i) - f_{k-1}(x_i)\|$	9.9920×10^{-14}
size(H)	120×120
max{eigenvalues(H)}	484.8513
min{eigenvalues(H)}	0.02472

Figures 26(a) and 26(c) show the autocorrelations corresponding to Fig. 25(a) and 25(c) respectively. Note that the autocorrelations in Fig. 26(a) and 26(c) look almost the same. There might be more than one signal that has the *exact* same autocorrelation. To convince the reader that this experiment does not fall into such a case of non-uniqueness, we show the difference of the two autocorrelations. The two images possess clear differences. Because the image is scaled for display, we also include the range of the values of the difference image in the caption. For the same purposes, we hereafter show difference images along with all autocorrelations.

Figures 27(a), 27(b), and 27(c) show the gradient images of the I-divergence computed at the true image, the initial estimate, and the final estimate, respectively. The gradient images are given to provide information on what the gradient looks like initially and how it has changed. However, since the images are scaled, they may be hard to compare if one is concerned with exact values. In such cases, one can refer to Tables 7, 8, and 9.

4.4.2.2 Experiment 2

Another interesting experiment was performed on this local minimum. We tried initializing the algorithm by a convex combination of the original image and the local minimum shown in Fig. 25(c):

$$f_{initial} = (1 - \alpha) f_{local} + \alpha f_{original}, \tag{45}$$

where $0 \leq \alpha \leq 1$. A small number 10^{-15} was added to the local minimum and the original image to make sure that all pixels would be allowed to take nonzero values and not be forced to always be zero.

Figure 28(c) shows the final estimate when $\alpha = 0.4$, and Fig. 28(b) shows what the initial image looks like. Even though this new estimate looks somewhat similar to the original image in Fig. 28(a), it shows distinctively different features on the edges. For example, some pixels near the center of the sector in the final estimate are erased out. Also, the two ends of the arc of the fan have lost their features. Figs. 29(a) and 29(c) show the autocorrelations of the images. Note that, although the original image and final estimate are quite different, the autocorrelations of these images are very similar. As before, to confirm that the new estimate is another local minimum, we tested the sufficient conditions and show some selected data from Exp. 2 in Table 7. Again, the I-divergence value computed at the final estimate is relatively large. The Hessian matrix is positive definite in terms of its eigenvalues. The associated gradient images are shown in Fig. 30.

Table 11: Selected data from Exp. 2.

Quantity	Value
$D[S, S]$	7.5830×10^{-13}
$D[S, R_{f_0}]$	716.3421
$D[S, R_{f_k}]$	6.3621
k	118113
$\max_i \lvert f_k(x_i) - f_{k-1}(x_i)\rvert$	9.9962×10^{-14}
size(H)	122×122
max{eigenvalues(H)}	493.0640
min{eigenvalues(H)}	0.0558

4.4.2.3 Experiment 3

Figure 31 shows another example of a local minimum. Figure 31(a) show the original true image consisting of a set of vertical lines to the right of a set of horizontal lines. Figure 31(c) shows the corresponding final estimate. Figure 32 shows the associated autocorrelations. Note that the autocorrelations are quite similar, but the original image and final estimate have interesting differences. There does not seem to be a definite space between the two sets of lines in the reconstruction; in addition, the lines near the border of the two line sets become "dotted." We do not have an intuitive explanation for this intriguing behavior. To confirm that the final estimate is a local minimum, we again perform the tests on the sufficient conditions. Table 8 contains information about Exp. 3. The related gradient images are shown in Fig. 33.

Table 12: Selected data from Exp. 3.

Quantity	Value
$D[S, S]$	8.9336×10^{-12}
$D[S, R_{f_0}]$	18021.1460
$D[S, R_{f_k}]$	45.0530
k	38210
$\max_i \lvert f_k(x_i) - f_{k-1}(x_i)\rvert$	9.9993×10^{-8}
size(H)	307×307
max{eigenvalues(H)}	1236.4256
min{eigenvalues(H)}	0.0327

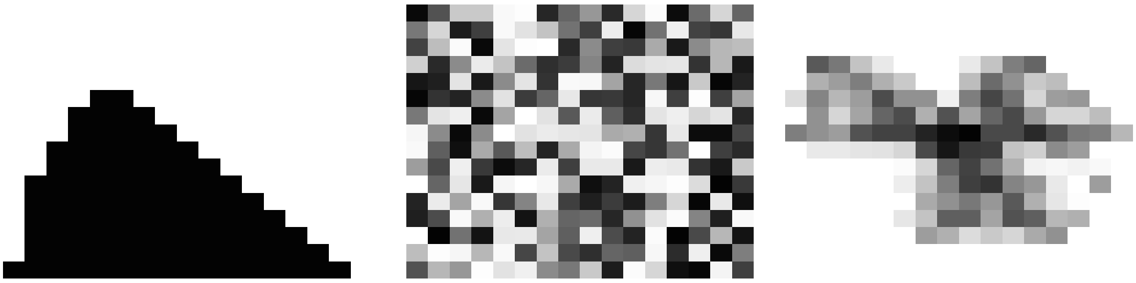

(a) (b) (c)

Figure 25: (a) Original image. (b) Initial estimate. (c) Final estimate.

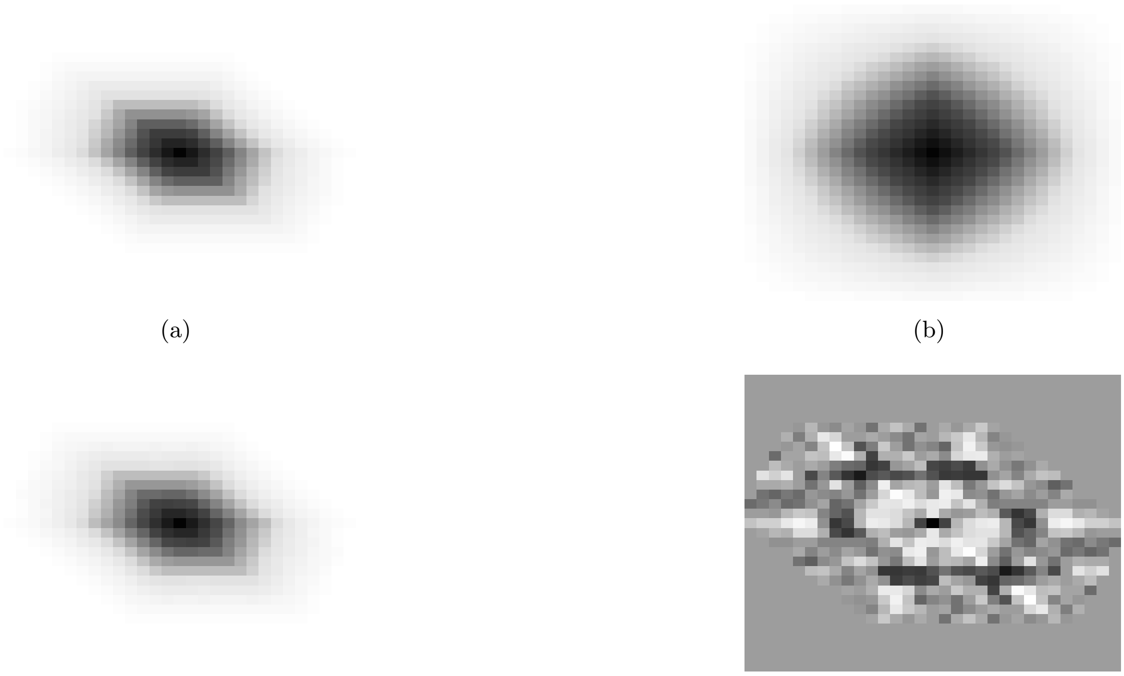

(a) (b) (c) (d)

Figure 26: Images associated with Fig. 25. (a) Autocorrelation of the original image. (b) Autocorrelation of the initial estimate. (c) Autocorrelation of the final estimate. (d) Difference of the autocorrelations in (a) and (c). The range of values is [-1.37 1.84].

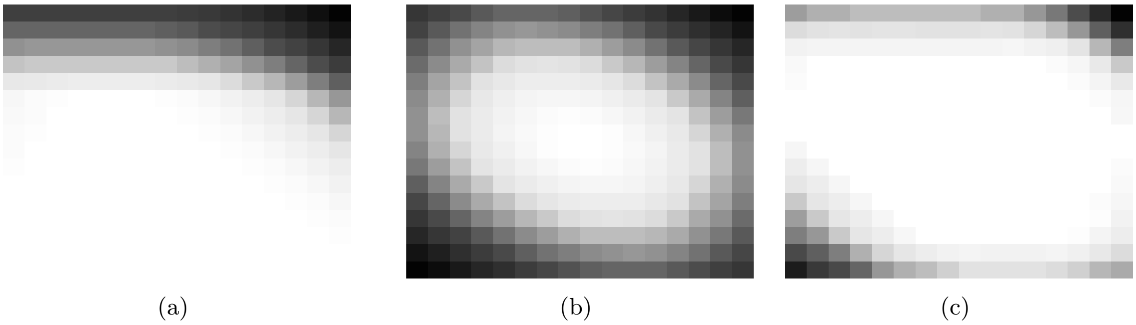


Figure 27: Images associated with Fig. 25. (a) Gradient of the original image. (b) Gradient of the initial estimate. (c) Gradient of the final estimate.

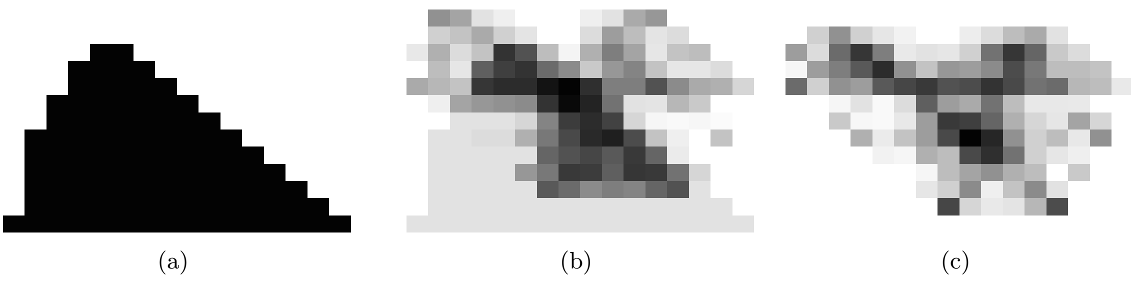


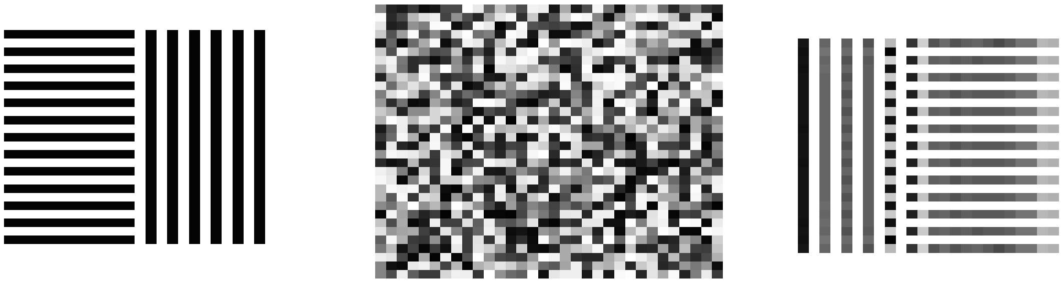
Figure 28: (a) Original image. (b) Initial estimate. (c) Final estimate.

(a) (b) (c) (d)

Figure 29: Images associated with Fig. 28. (a) Autocorrelation of the original image. (b) Autocorrelation of the initial estimate. (c) Autocorrelation of the final estimate. (d) Difference of the autocorrelations in (a) and (c). The range of values is [-1.39 2.64].

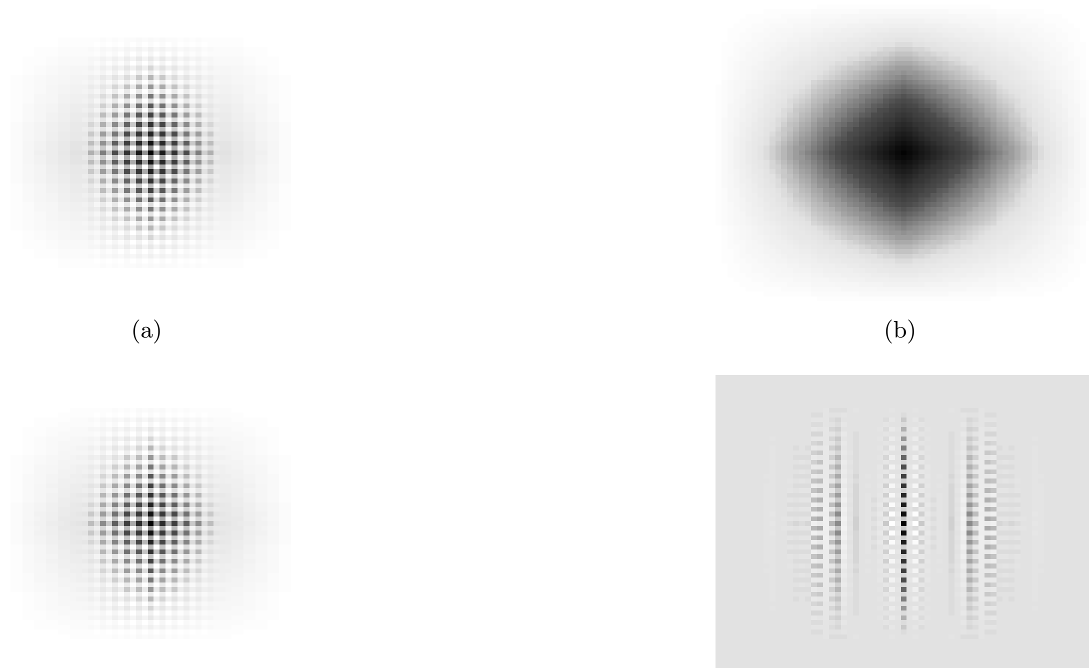
(a) (b) (c)

Figure 30: Images associated with Fig. 28. (a) Gradient of the original image. (b) Gradient of the initial estimate. (c) Gradient of the final estimate.



(a) (b) (c)

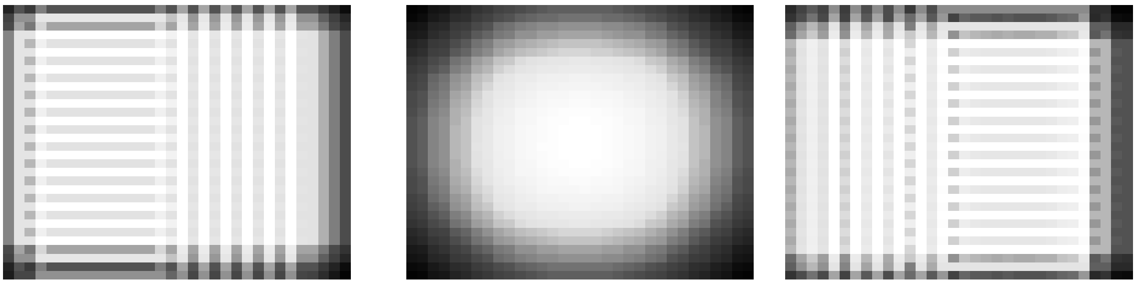
Figure 31: (a) Original image. (b) Initial estimate. (c) Final estimate.



(a) (b)

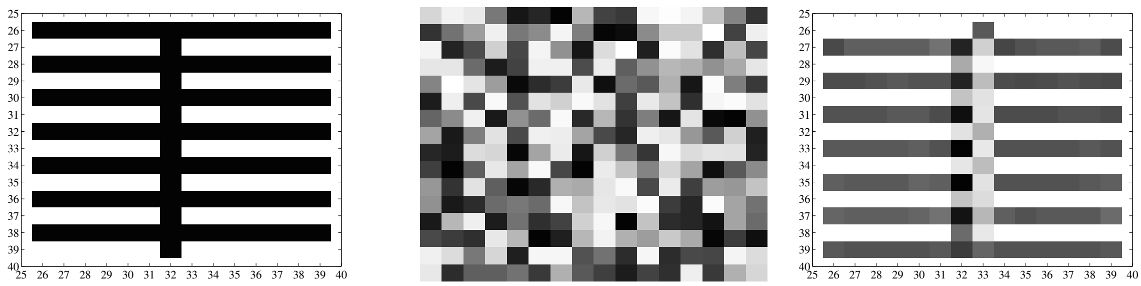
(c) (d)

Figure 32: Images associated with Fig. 31. (a) Autocorrelation of the original image. (b) Autocorrelation of the initial estimate. (c) Autocorrelation of the final estimate. (d) Difference of the autocorrelations in (a) and (c). The range of values is [-7.56 2.19].



(a) (b) (c)

Figure 33: Images associated with Fig. 31. (a) Gradient of the original image. (b) Gradient of the initial estimate. (c) Gradient of the final estimate.



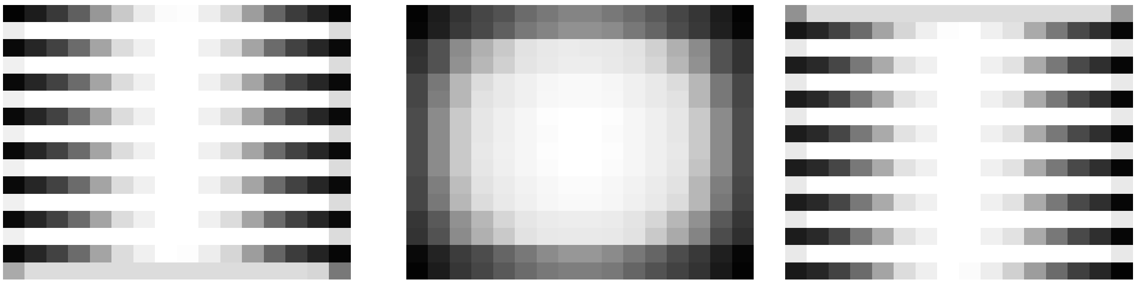
(a) (b) (c)

Figure 34: (a) Original image. (b) Initial estimate. (c) Final estimate.

(a) (b)

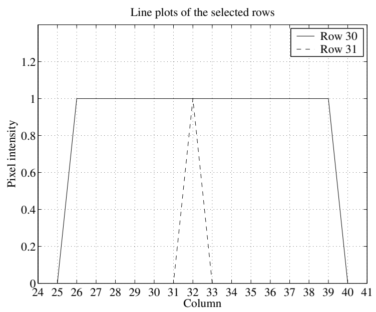
(c) (d)

Figure 35: Images associated with Fig. 34. (a) Autocorrelation of the original image. (b) Autocorrelation of the initial estimate. (c) Autocorrelation of the final estimate. (d) Difference of the autocorrelations in (a) and (c). The range of values is [-0.58 0.91].

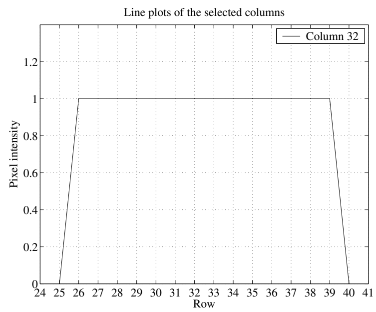


(a) (b) (c)

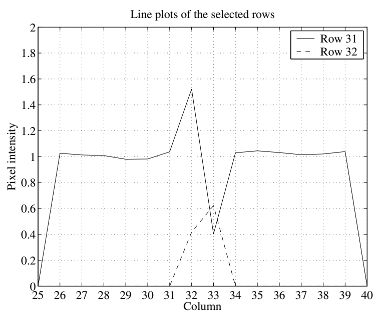
Figure 36: Images associated with Fig. 34. (a) Gradient of the original image. (b) Gradient of the initial estimate. (c) Gradient of the final estimate.



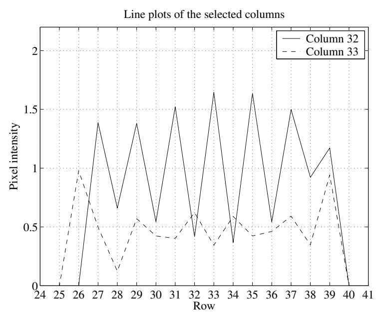
(a)



(b)

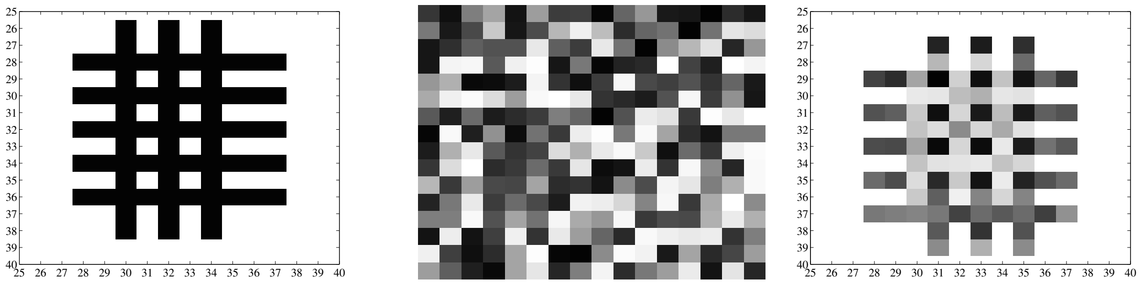


(c)



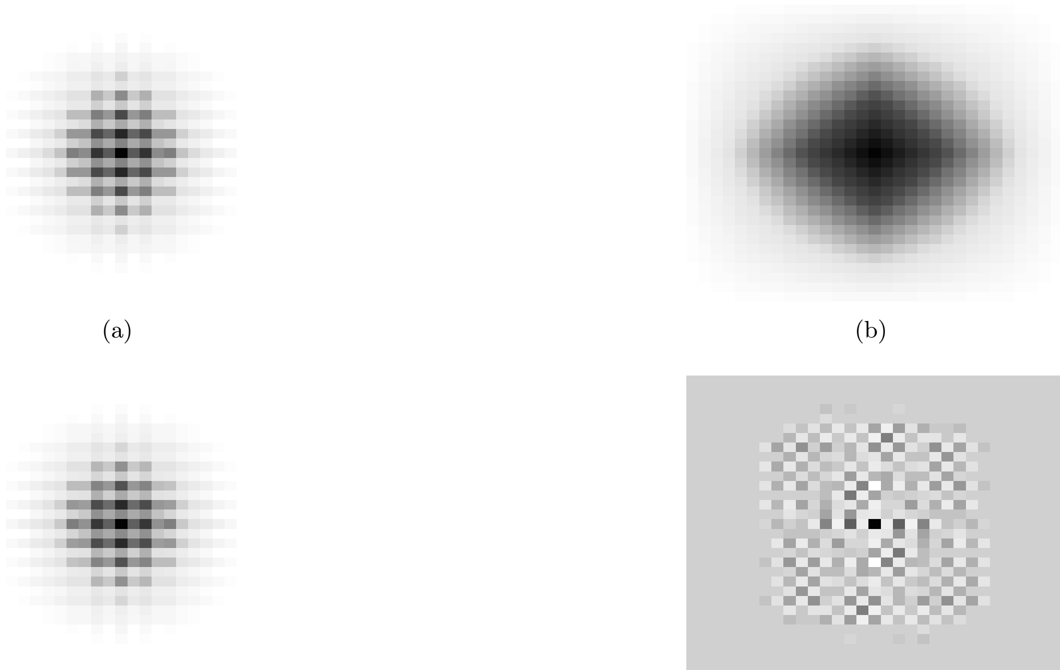
(d)

Figure 37: Line plots associated with Fig. 34 are shown. The rows and columns are selected such that all other lines are common in the overall trend with one of the rows or columns. (a) Line plots of some selective rows of the original image. (b) Line plots of some selective columns of the original image. (c) Line plots of some selective rows of the final estimate. (d) Line plots of some selective columns of the final estimate.



(a) (b) (c)

Figure 38: (a) Original image. (b) Initial estimate. (c) Final estimate.



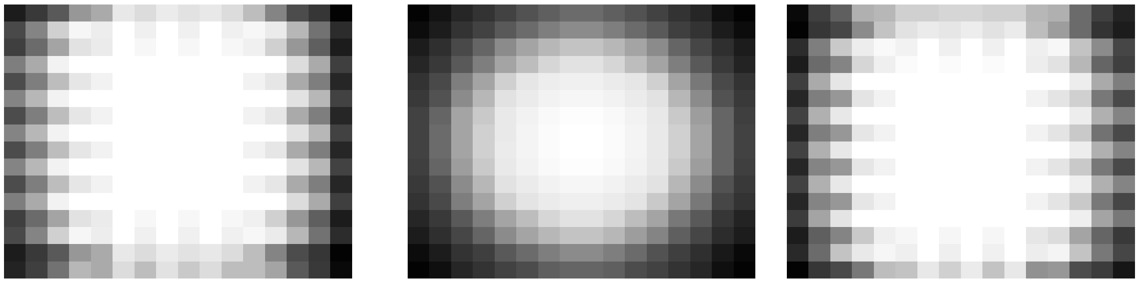
(a) (b)

(c) (d)

Figure 39: Images associated with Fig. 38. (a) Autocorrelation of the original image. (b) Autocorrelation of the initial estimate. (c) Autocorrelation of the final estimate. (d) Difference of the autocorrelations in (a) and (c). The range of values is [-0.80 1.80].

4.4.3 Mild Artifacts

The experiments described in the rest of this section were originally performed to try to analyze the resolution of the Schulz-Snyder algorithm. As our research progressed, it became apparent that resolution in the usual sense was not necessarily an issue (at least in the noiseless cases considered in this chapter); points and lines are often reconstructed without



(a) (b) (c)

Figure 40: Images associated with Fig. 38. (a) Gradient of the original image. (b) Gradient of the initial estimate. (c) Gradient of the final estimate.

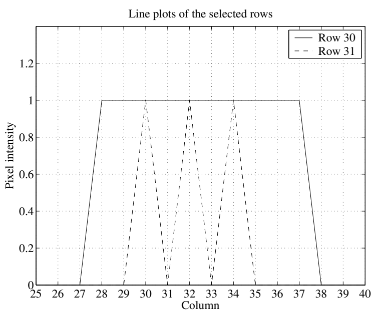
any apparent blurring. However, two adjacent lines may sometimes appear blurred together. Sometimes, artifacts in such cases may have apparent tendencies as in the experiments to be discussed in Section 4.4.4. Nonetheless, these mild artifacts, in most cases, are quite unpredictable.

4.4.3.1 Experiment 4

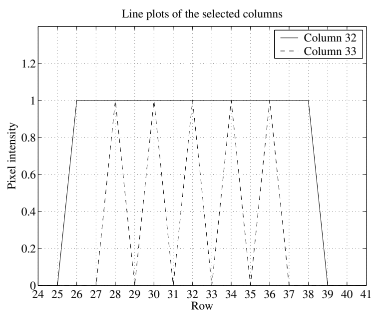
Figure 34(a) shows a vertical line superimposed on seven alternating horizontal lines. Figure 34(c) shows the corresponding final estimate. The estimate shows a somewhat weird reconstruction; the vertical line is spread to the left, except for the protruding part. Figure 37 shows line plots of some selected rows and columns. As we may see in Figs. 37(c) and 37(d), the ways that the vertical lines are reconstructed are quite different for the two different columns, and Row 31 is also reconstructed strangely. On the other hand, the gradients and autocorrelations of the original image and final estimate look very similar as shown in Figs. 35 and 36. As before, the sufficient conditions were tested on the final estimate to see if it has converged. Related data are given in Table 9.

4.4.3.2 Experiment 5

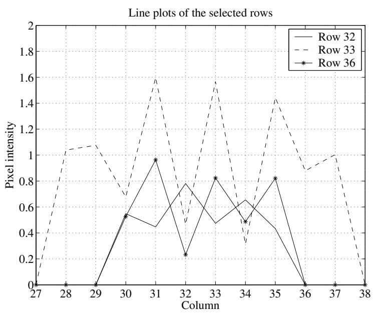
The unpredictability of such phenomena becomes more obvious with three vertical lines spaced by one pixel superimposed on five horizontal lines. Figure 38(a) shows the original true image, and Fig. 38(c) shows the final estimate of the original image. The reason that we remove a few horizontal lines is to emphasize the effects of the vertical line reconstruction more than the horizontal line reconstruction. Note that, in the final estimate, we obtain a 6



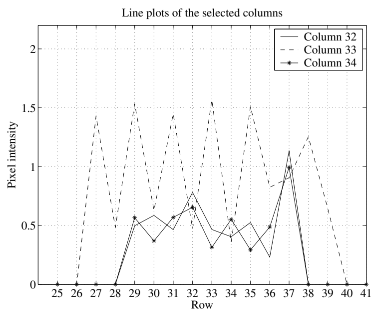
(a)



(b)



(c)



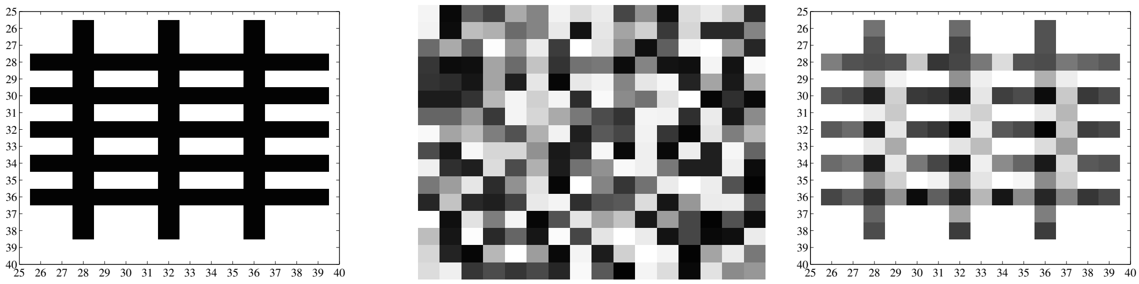
(d)

Figure 41: Line plots associated with Fig. 38 are shown. The rows and columns are selected such that all other lines are common in the overall trend with one of the rows or columns. (a) Line plots of some selected rows of the original image. (b) Line plots of some selected columns of the original image. (c) Line plots of some selected rows of the final estimate. (d) Line plots of some selected columns of the final estimate.

$\times$ 6 block that looks like a checkerboard. However, pixel values in the block are somewhat randomly distributed. Figure 41 supports this observation. In this case, the corresponding autocorrelations still look very similar, but the gradients show a small difference; compare the top and bottom rows of Figs 40(a) and 40(c). The autocorrelations for this experiment are shown in Fig. 39. Table 14 shows the related data illustrating that the final estimate satisfies the sufficient conditions and hence is a local minimum.

4.4.3.3 Experiment 6

Experiment 6 illustrates a local minima that bears an overall resemblance to the correct answer, but with some unusual minor artifacts. Figure 42(a) shows the original true image consisting of three vertical lines spaced by three pixels superimposed on five alternating lines. Figure 42(c) is the final estimate of the original pattern. By carefully observing Row



(a) (b) (c)

Figure 42: (a) Original image. (b) Initial estimate. (c) Final estimate.

Table 13: Selected data from Exp. 4.

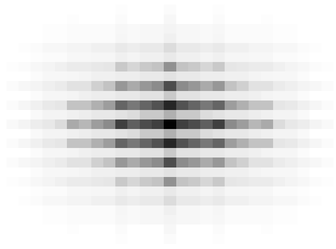
Quantity	Value
$D[S, S]$	6.2433×10^{-13}
$D[S, R_{f_0}]$	1581.9542
$D[S, R_{f_k}]$	0.1572
k	57720
$\max_i \lvert f_k(x_i) - f_{k-1}(x_i)\rvert$	9.9997×10^{-8}
size(H)	111×111
max{eigenvalues(H)}	469.1720
min{eigenvalues(H)}	0.0574

35, we are able to see the vertical lines are spread among more than two bins. Figures 45(c) and 45(d) show line plots of some interesting rows and columns in addition to Row 35. The line plots shown in the figures are fairly different than the line plots in Figs. 41(c) and 41(d). Besides, the columns in Fig. 45(d) have a less predictable pattern than seen in, for instance, Fig. 49(d).

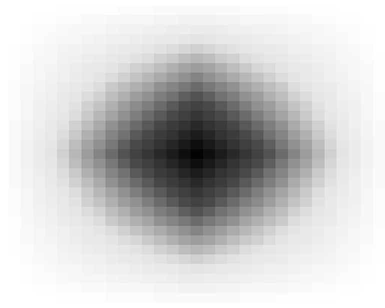
Interestingly, the corresponding gradients look quite similar, unlike in Exp. 5. Also, the corresponding autocorrelations look quite similar, as expected, and are shown in Fig. 43. As before, Table 15 shows related data from Exp. 6 concerning the testing of the sufficient conditions.

4.4.4 Straddle Loss Effects

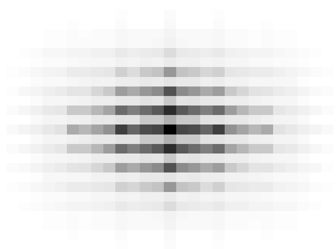
Unlike the experiments in the latter part of Section 4.4.3, the experiments discussed in this section show consistent behaviors.



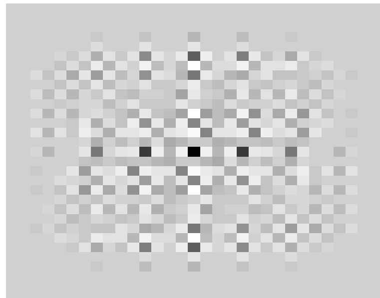
(a)



(b)



(c)



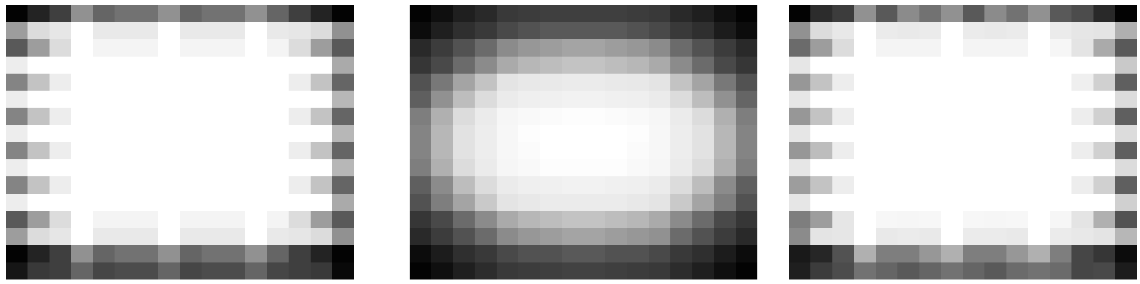
(d)

Figure 43: Images associated with Fig. 42. (a) Autocorrelation of the original image. (b) Autocorrelation of the initial estimate. (c) Autocorrelation of the final estimate. (d) Difference of the autocorrelations in (a) and (c). The range of values is [-0.93 2.10].

4.4.4.1 Experiment 7

Figure 46(a) shows the original image of a vertical line superimposed on seven horizontal lines, but without any protruding parts as in Fig. 34(a). Figure 46(c) shows the final estimate of the original image. As can be seen, all the alternating rows (*e.g.*, Rows 26 and 28, or Rows 27 and 29) show similar tendencies, even though the values are fluctuating about 1 in the reconstructed horizontal lines. The two reconstructed columns also show a similar tendency. In contrast to the reconstructions of the columns in Exps. 4, 5, and 6, the columns reconstructed in this experiment have a somewhat clear pattern (compare Column 33 in Figs. 37(d) and 49(d)). Note that the reconstructions of the vertical lines are spread between only two bins. In this experiment, both the autocorrelations and gradients of the original image and final estimate look extremely close (see Figs 47 and 48). The data showing that the final estimate is a local minimum are given in Table 16.

We believe the spreading is due to a *straddle loss* effect such as that seen when one



(a) (b) (c)

Figure 44: Images associated with Fig. 42. (a) Gradient of the original image. (b) Gradient of the initial estimate. (c) Gradient of the final estimate.

Table 14: Selected data from Exp. 5.

Quantity	Value
$D[S,S]$	5.900×10^{-13}
$D[S,R_{f_0}]$	1645.0375
$D[S,R_{f_k}]$	1.0190
k	15108
$\max_i \lvert f_k(x_i) - f_{k-1}(x_i)\rvert$	9.9979×10^{-8}
size(H)	86×86
max{eigenvalues(H)}	353.1094
min{eigenvalues(H)}	0.1541

applies an FFT to a sinusoid whose frequency does not fall exactly on one of the FFT bins, and the energy from that sinusoid is split between two adjacent bins (see, for instance, Ref. [81, p. 165]).

4.4.4.2 Experiment 8

We performed another experiment with a pattern of two vertical lines superimposed on seven horizontal lines. The original image is shown in Fig. 50(a), and the final estimate is shown in Fig. 50(c). Table 17 supports the fact that the final estimate is a local minimum based on the given data. The reconstructions of the vertical lines are spread between only two bins as in Exp. 7, which is a clear straddle loss effect. This is obviously different than the effects in Exp. 6, whose original image is also composed of two vertical lines with some alternating horizontal lines. The line plots shown in Fig. 53 show consistent patterns and manifest the straddle loss effects along Row 30 (see Figs. 53(c) and 53(d)).

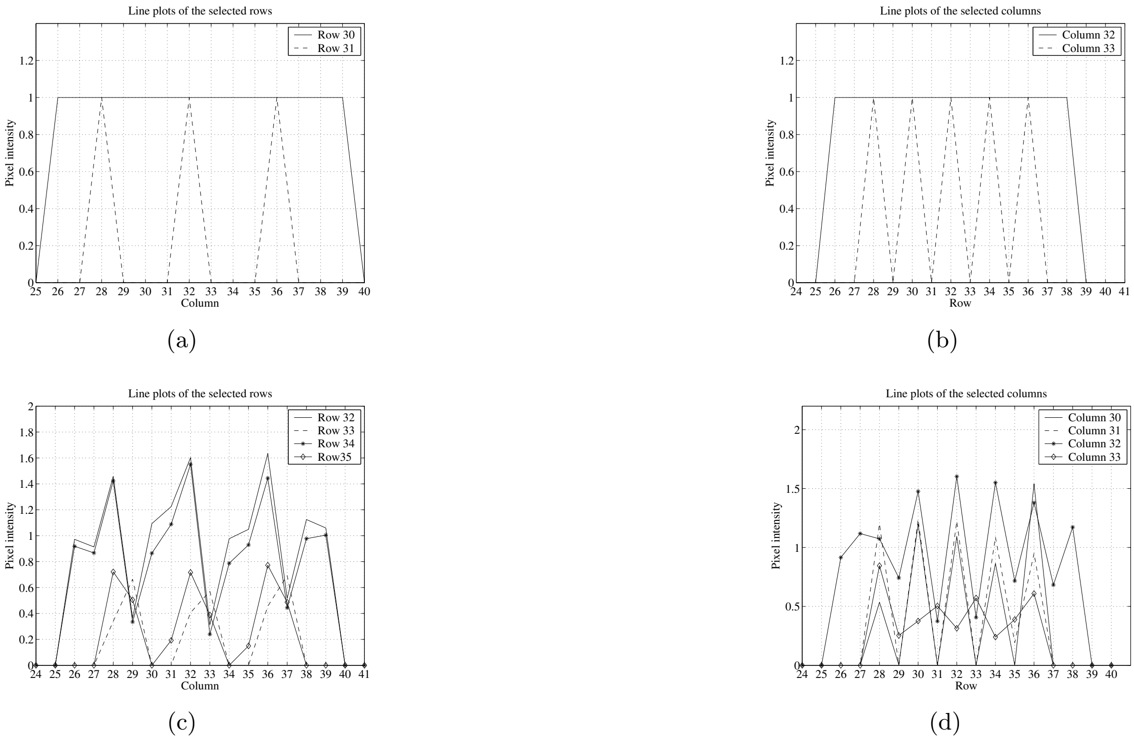


Figure 45: Line plots associated with Fig. 42 are shown. The rows and columns are selected such that all other lines are common, in the overall trend, with one of the rows or columns. (a) Line plots of some selected rows of the original image. (b) Line plots of some selected columns of the original image. (c) Line plots of some selected rows of the final estimate. (d) Line plots of some selected columns of the final estimate.

The autocorrelations and gradients are shown in Figs. 51 and 52, respectively.

Recall that translating and/or mirroring an image does not change its autocorrelation function; hence, an algorithm trying to reconstruct an image from its autocorrelation may reconstruct a shifted version of the original image. In fact, the I-divergence surface has many global minima, some of which correspond to versions of the image shifted by an integer number of pixels. We suspect that in the above experiments, where a small amount of directional blurring is observed, the algorithm is trying to reconstruct the image translated a half-pixel to the right of its original location, resulting in the straddle loss effect as the algorithm tries to put the vertical line in between pixel values. The most surprising aspect of these experiments is that no straddle loss effects were observed for the repeated horizontal lines (and similarly for the cases with repeated vertical lines, which are not shown); we do not have an explanation for this intriguing behavior at present.

Table 15: **Selected data from Exp. 6.**

Quantity	Value
$D[S,S]$	5.0976×10^{-13}
$D[S,R_{f_0}]$	1982.5427
$D[S,R_{f_k}]$	1.2542
k	101158
$\max_i \lvert f_k(x_i) - f_{k-1}(x_i)\rvert$	9.9999×10^{-9}
size(H)	110×110
max{eigenvalues(H)}	454.0213
min{eigenvalues(H)}	0.0543

Table 16: **Selected data from Exp. 7.**

Quantity	Value
$D[S,S]$	8.5242×10^{-13}
$D[S,R_{f_0}]$	3716.3495
$D[S,R_{f_k}]$	0.0807
k	94240
$\max_i \lvert f_k(x_i) - f_{k-1}(x_i)\rvert$	9.9999×10^{-8}
size(H)	110×110
max{eigenvalues(H)}	474.7070
min{eigenvalues(H)}	0.0032

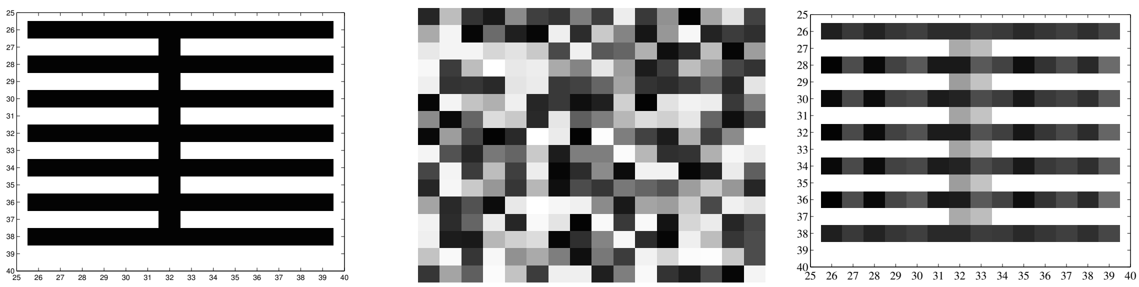
4.4.4.3 Experiment 9

In a final experiment, we initialized the algorithm with the true image with a constant ϵ added to every pixel in a rectangle surrounding the true image, where the true image is taken to be the pattern in Fig. 50(a) from Experiment 8. The support of the constant rectangle is the same as the support for the initial uniform estimate used in the experiment that resulted in Fig. 50(c). Using a small value of $\epsilon = 10^{-6}$, the algorithm converges to the true answer without any blurring, in the sense that the vertical lines have value 1 and the valleys are practically zero. When ϵ is boosted to 0.5, the straddle loss effect is again observed; however, this time, the two vertical lines are quite distinguishable. The value of two peaks in the estimated image is 0.8, and the value of two valleys is 0.2; this can be thought of as an answer lying between the correct result and the block result in Fig. 50(c). To save space, we do not show the plots for this experiment since they provide little

Table 17: Selected data from Exp. 8.

Quantity	Value
$D[S,S]$	5.7178×10^{-13}
$D[S,R_{f_0}]$	2685.6983
$D[S,R_{f_k}]$	0.0292
k	257827
$\max_i \lvert f_k(x_i) - f_{k-1}(x_i)\rvert$	9.9999×10^{-9}
size(H)	122×122
max{eigenvalues(H)}	522.9897
min{eigenvalues(H)}	0.0022

information beyond the textual description and are largely redundant with the preceding plots.



(a) (b) (c)

Figure 46: (a) Original image. (b) Initial estimate. (c) Final estimate.

4.5 Conclusions

We have derived the second derivatives of the I-divergence between a measured autocorrelation and an estimated autocorrelation in terms of the image estimate, and used that to illustrate that the Schulz-Snyder algorithm may converge to local minima of the I-divergence surface. In some cases, the local minima correspond to shifts of the true image. Sometimes this shift may consist of a fraction of a pixel width. Such local minima may be thought of as "close" to a true global minimum, and are not that objectionable. In other cases, the local minima may bear little resemblance to the true object.

The Schulz-Snyder algorithm bears a striking resemblance to the Richardson-Lucy [72]

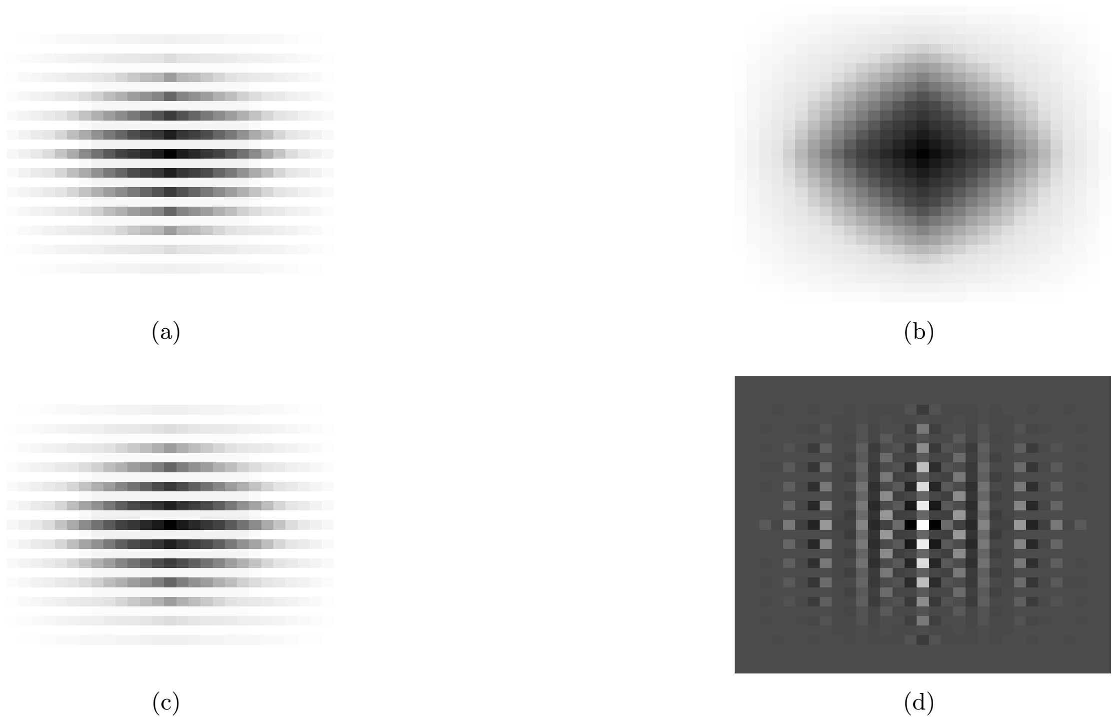
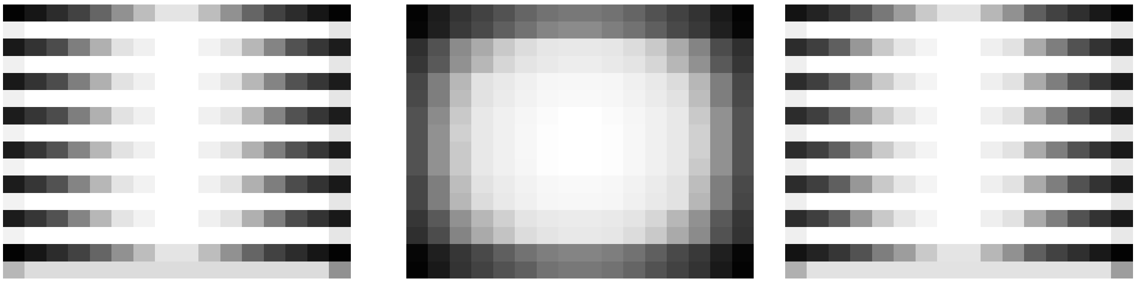


Figure 47: Images associated with Fig. 46. (a) Autocorrelation of the original image. (b) Autocorrelation of the initial estimate. (c) Autocorrelation of the final estimate. (d) Difference of the autocorrelations in (a) and (c). The range of values is [-0.70 0.39].

algorithm for reconstructing an image from a blurred version of that image. The Richardson-Lucy algorithm can be shown to minimize the I-divergence between the measured blurred image and a blurred version of the estimated image [107]. In the Richardson-Lucy case, one can show that the objective function has a unique global minimum [107]. We have shown that the Schulz-Snyder scenario is far more complex. Not only are the distinct global minima due to the invariance of the autocorrelation function to shift, but there may also be numerous local minima in which the algorithm may become trapped. To our knowledge, this work is the first to report these effects concerning the Schulz-Snyder algorithm.

Modifications to the Schulz-Snyder algorithm must be found which will allow it to break out of local minima. In our experiments, we had the advantage of knowing what the true images were; in practice, it may be difficult to tell if a found minimum is a local minimum or one of the true global minima. The Schulz-Snyder algorithm is promising, since even without these modifications, it has still been shown to converge in many situations. Given



(a) (b) (c)

Figure 48: Images associated with Fig. 46. (a) Gradient of the original image. (b) Gradient of the initial estimate. (c) Gradient of the final estimate.

this promise, we hope this work will inspire the phase retrieval community to explore this path.

We were drawn to the Schulz-Snyder iteration due to its analogy with multiplicative deblurring iterations that are popular within the optics community, such as the Richardson-Lucy iteration. A useful avenue for future work would be to compare the Schulz-Snyder iteration with some other traditional optimization techniques such as Newton's methods, trust region methods, and conjugate gradient methods.

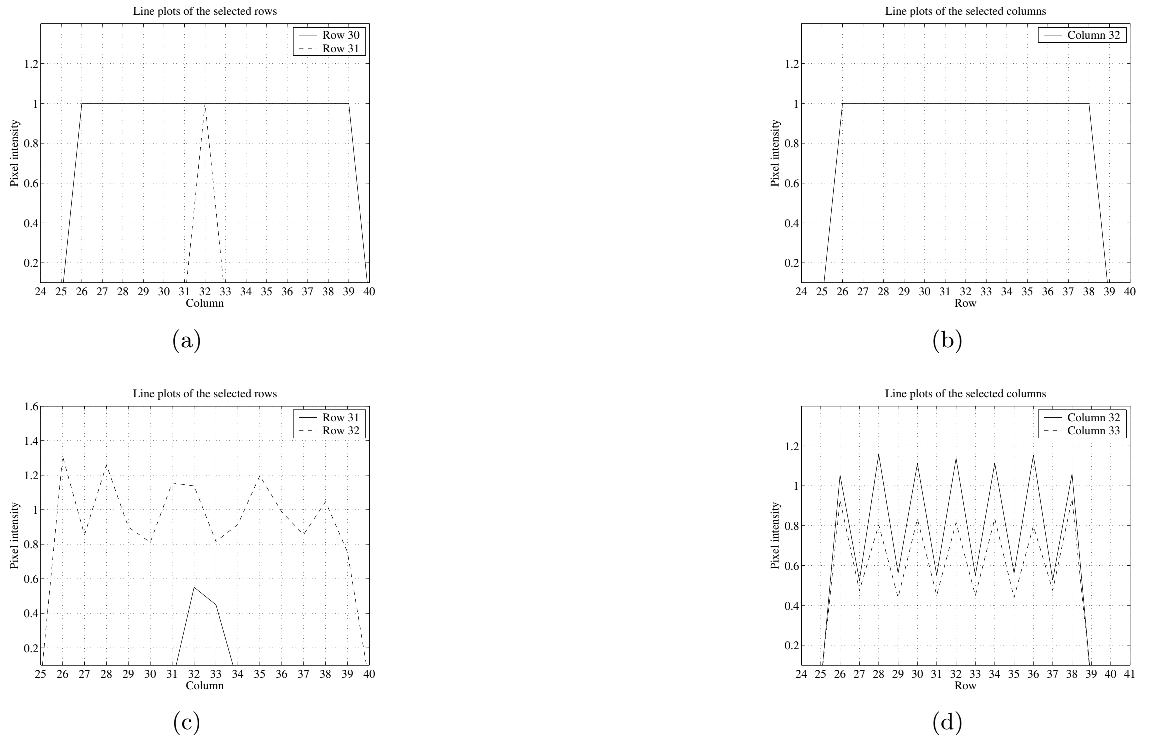


Figure 49: Line plots associated with Fig. 46 are shown. The rows and columns are selected such that all other lines are common in the overall trend with one of the rows or columns. (a) Line plots of some selective rows of the original image. (b) Line plots of some selective columns of the original image. (c) Line plots of some selective rows of the final estimate. (d) Line plots of some selective columns of the final estimate.

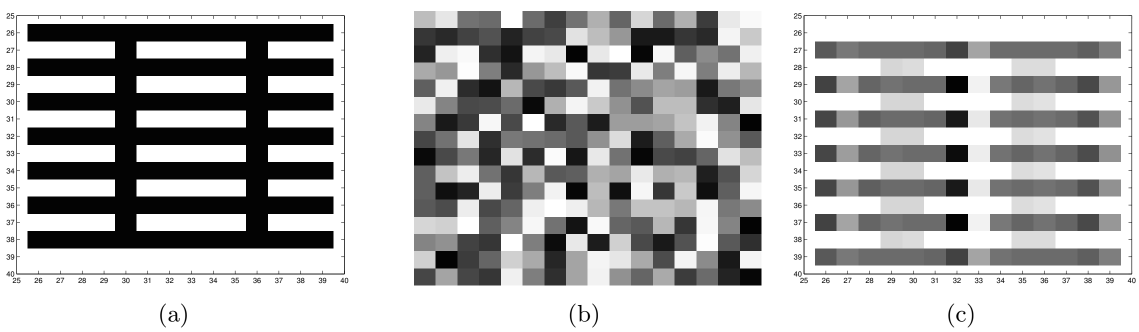
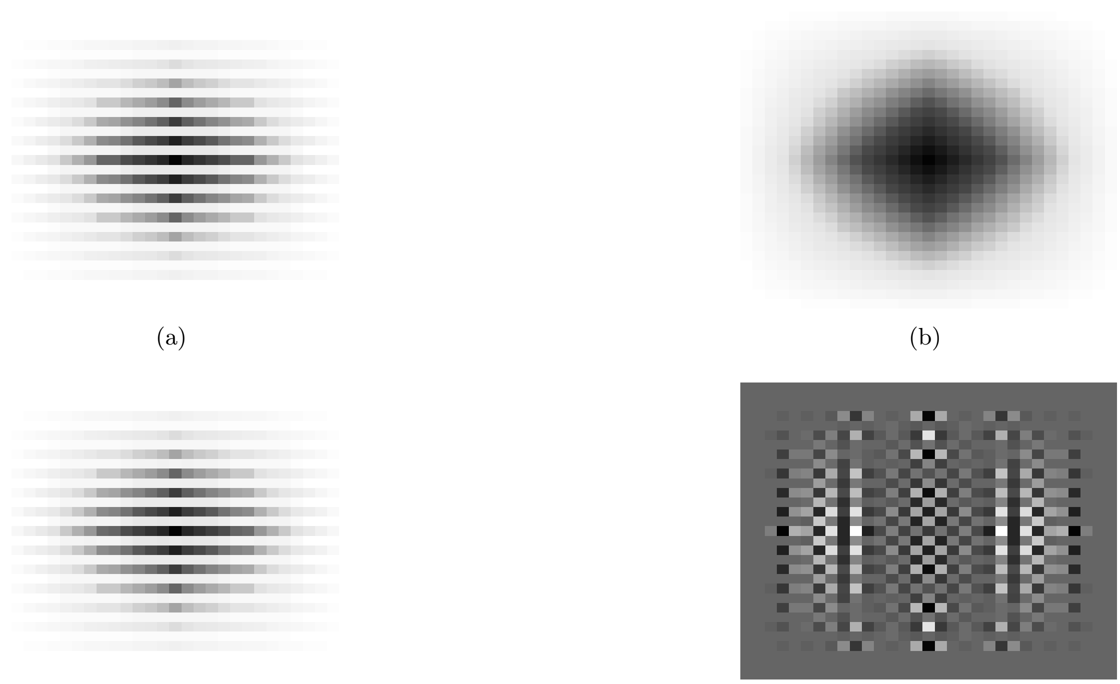
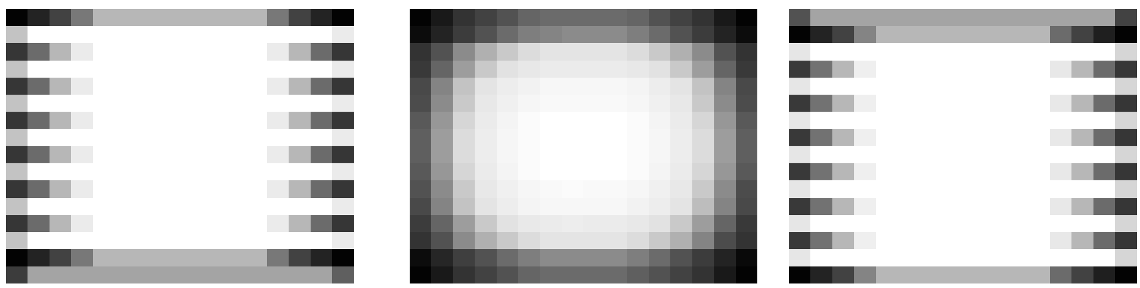


Figure 50: (a) Original image. (b) Initial estimate. (c) Final estimate.



(a) (b) (c) (d)

Figure 51: Images associated with Fig. 50. (a) Autocorrelation of the original image. (b) Autocorrelation of the initial estimate. (c) Autocorrelation of the final estimate. (d) Difference of the autocorrelations in (a) and (c). The range of values is [-0.20 0.20].



(a) (b) (c)

Figure 52: Images associated with Fig. 50. (a) Gradient of the original image. (b) Gradient of the initial estimate. (c) Gradient of the final estimate.

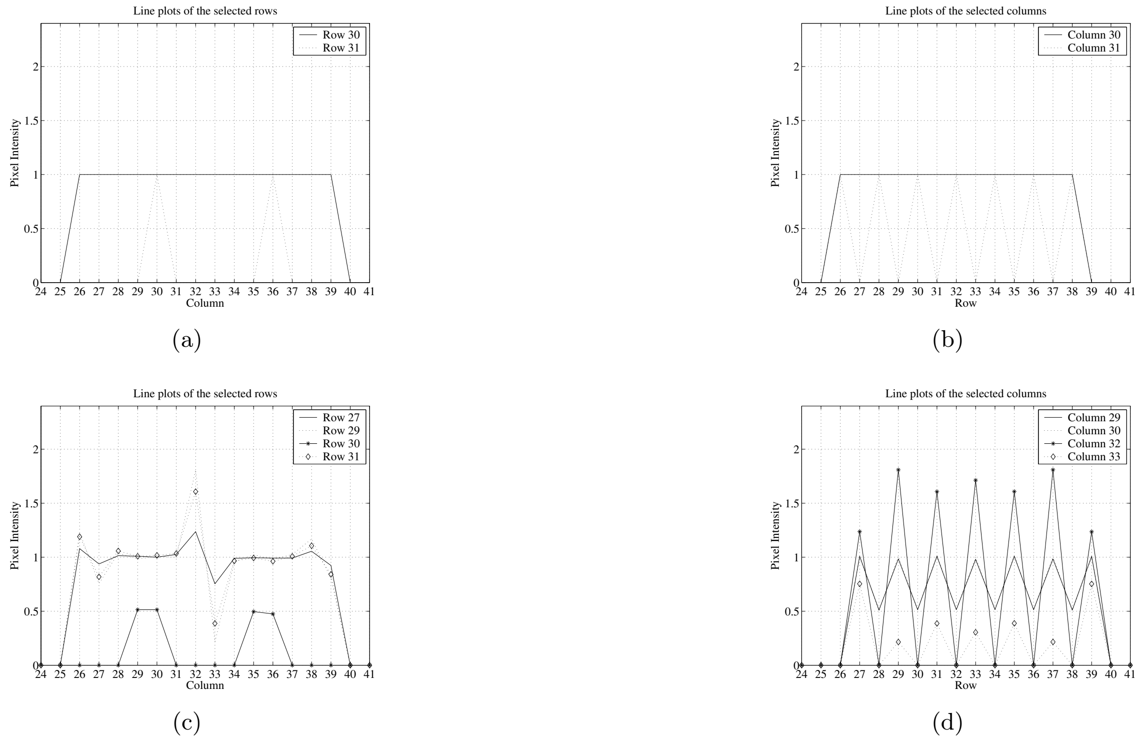


Figure 53: Line plots associated with Fig. 50 are shown. The rows and columns are selected such that all other lines are common, in the overall trend, with one of the rows or columns. (a) Line plots of some selected rows of the original image. (b) Line plots of some selected columns of the original image. (c) Line plots of some selected rows of the final estimate. (d) Line plots of some selected columns of the final estimate.

CHAPTER V

PHASE RETRIEVAL FROM NOISY DATA BASED ON MINIMIZATION OF PENALIZED I-DIVERGENCE

5.1 Introduction

In various scientific applications such as astronomical imaging through extreme atmospheric turbulence and x-ray crystallography, it is impossible to directly observe the objects of interest with current technology. In some problems, Fourier magnitudes – but not Fourier phases – are obtained. For example, in crystallography, we want to find the interatomic structure of a molecule, but the structure cannot be directly observed with any practical devices because of physical limitations [77,91,116]. Instead, we can obtain Fourier magnitudes by shooting x-rays through a crystal, but Fourier phase information is entirely lost. In astronomical imaging, we can directly obtain the autocorrelation of an object through photon differencing [92, 93]. However, the autocorrelation is the inverse Fourier transform of the squared Fourier magnitudes of the object, meaning that the Fourier phases are completely lost.

As in these two applications, phase retrieval can be approached from two viewpoints. Of course, the first approach is to retrieve Fourier phases from the corresponding Fourier magnitudes, as in x-ray crystallography. The other approach is to estimate a function from its autocorrelation, as in astronomical imaging. Note that the knowledge of a function is equivalent to the knowledge of its Fourier magnitudes and phases.

Based on the latter idea, Schulz and Snyder found an Expectation-Maximization (EM) algorithm that attempts to recover a function from its autocorrelation for their astronomical imaging application [92]. In their underlying stochastic model, the data are assumed to follow a Poisson point process; the EM algorithm maximizes the corresponding Poisson likelihood. Furthermore, they noted that their EM algorithm, in its asymptotic form, which we call the Schulz-Snyder algorithm (see Chapter 4), produces a sequence of estimates that can minimize an information-theoretic discrepancy measure called Csiszár's I-divergence (also called cross entropy in the related literature [10–12, 74]). The asymptotic form is obtained by assuming an infinite number of data samples and by using the weak law of large numbers [92]. (Similar arguments were made earlier by Snyder *et al.* [105] in the context of emission tomography, which also assumes a Poisson data model.) Later, they formally derived the Schulz-Snyder algorithm [93] by minimizing the I-divergence via the Kuhn-Tucker conditions [73]. Although the papers by Schulz and Snyder gave a few numerical examples, none of their experiments involved noise. To our knowledge, this work is the first to explore the effect of noise on the Schulz-Snyder iteration.

Csiszár's I-divergence [23] is an information-theoretic discrepancy measure defined on two nonnegative functions. It may be thought of as a generalization of the Kullback-Leibler distance [62,64]. An important result of Csiszár's work [23] is that if the functions involved in an inverse problem are nonnegative, minimizing the I-divergence measure is the only method consistent with a set of intuitive postulates such as regularity and locality, which are desirable for estimation problems. Methods of minimizing the I-divergence have been popular in various estimation applications [10–12, 107, 114].

In general, maximum-likelihood estimates (MLE) are highly inclined to become rough when data are noisy. Estimates for phase retrieval problems, obtained by minimizing the I-divergence between the measured autocorrelation and the autocorrelation of an estimate, are equivalent to MLE under a Poisson autocorrelation data model. Therefore, it is interesting to investigate what impact noise can have on minimum I-divergence estimates in phase retrieval. In both astronomical imaging and x-ray crystallography, noise may be modeled by Poisson random processes [92, 95]. In particular, x-ray crystallography data are usually measured with charge-coupled-device (CCD) cameras, whose detectors' readout noise can be modeled by a Poisson random process [100, 101].

Good's roughness penalty [39] has been known to be helpful for reducing the noise artifacts in maximum-likelihood estimation for several applications including emission tomography [78, 102, 103] and optical sectioning microscopy [56]. The penalty encourages smooth estimates by penalizing the differences between an estimate and shifted versions of the estimate. Since minimum I-divergence estimates for phase retrieval from noisy data are also rough, we study effects of Good's roughness on the estimates. A particularly nice aspect about Good's roughness penalty is that it can interpreted in terms of I-divergences. This supplies insights on how Good's roughness operates on estimates.

Although Good's roughness can reasonably suppress noise artifacts, it tends to smear edges, which is often disturbing. The total variation (TV) penalty has been known to provide estimates that smooth noise while preserving the edges. [13, 14, 19, 20, 69, 90]. This motivates us to study effects of the TV penalty on minimum I-divergence estimates for phase retrieval. The TV penalty has also been used in emission tomography [55].

In a sense, phase retrieval can be viewed as a blind deconvolution problem, where the unknown kernel is a reflection of the object being imaged. The TV penalty has found some success in regularizing estimates (as well as unknown kernels) in blind deconvolution [15–17]. This serves as another motivation for considering the TV penalty in our study.

When regularizing minimum I-divergence estimates by Good's roughness, or TV penalties, the pertinent optimization problem that needs to be solved at each iteration becomes complicated because the components of estimates are "coupled" by the penalties. Green's one-step-late (OSL) algorithms [41, 42] are techniques proposed to easily resolve such issues in EM algorithms. Based on the important theoretical fact that minimum I-divergence algorithms are asymptotically equivalent to certain types of EM algorithms under Poisson data models, we adapt Green's OSL for use in our phase retrieval algorithms.

This chapter is organized as follows. Section 5.2 discusses unconstrained phase retrieval algorithms based on minimizing Csiszár's I-divergence. We discuss penalties in detail and derive our constrained algorithms for phase retrieval using I-divergence in Sec. 5.3. Section 5.4 illustrates and discusses interesting experiments. Finally, we conclude our study in Sec. 5.5.

5.2 *Unconstrained Phase Retrieval Algorithms*

5.2.1 An Algorithm for Unaliased Autocorrelations: The Schulz-Snyder Algorithm

In astronomical imaging, the autocorrelation of an object, rather than its Fourier magnitudes, is directly obtained via manipulation of measured data [92]. This autocorrelation is "continuous" in that the associated Fourier magnitudes, which yield an "unaliased" autocorrelation, are not undersampled as in x-ray crystallography [77].

Schulz and Snyder found an iterative algorithm for recovering nonnegative functions from *n-th* order correlations [93]. Here, we are interested specifically in the $n = 2$ case of recovery from "unaliased" autocorrelations, which is equivalent to phase retrieval. For implementation on a computer, we discretize all functions of interest. The algorithm estimates functions from their autocorrelations by minimizing Csiszár's I-divergence:

$$I(S||R_\lambda) = \sum_{\mathbf{y}} \left\{ S(\mathbf{y}) \log \frac{S(\mathbf{y})}{R_\lambda(\mathbf{y})} + R_\lambda(\mathbf{y}) - S(\mathbf{y}) \right\}, \tag{46}$$

where $S = R_f$ is the autocorrelation of some true but unknown f that we desire to estimate from S, and the autocorrelation of an estimate λ is defined as

$$R_\lambda(\mathbf{y}) = \sum_{\mathbf{x}} \lambda(\mathbf{x})\lambda(\mathbf{x}+\mathbf{y}), \tag{47}$$

where $\mathbf{x} \in \mathcal{X}$ and $\mathbf{y} \in \mathcal{Y}$; $\mathcal{X} = \{1, 2, \ldots, N\} \times \{1, 2, \ldots, M\}$, and

$$\mathcal{Y} \stackrel{def}{=} \{\mathbf{y} : \mathbf{y} = \mathbf{x}_1 - \mathbf{x}_2, (\mathbf{x}_1, \mathbf{x}_2) \in \mathcal{X}^2\}. \tag{48}$$

The algorithm attempts to minimize the following objective function $J(\lambda) = I(S||R_\lambda)$ subject to the constraints

$$\begin{aligned} C(\lambda) = \sum_{\mathbf{x}} \lambda(\mathbf{x}) \ &= \ C(f) \\ \lambda \geq 0, & \end{aligned} \tag{49}$$

where $\{C(\lambda)\}^2 = \sum_{\mathbf{y}} R_\lambda(\mathbf{y})$ (see Property 3.4 in [93, p. 1269]). Note that $C(f)$ can be obtained even if f is unknown.

The Schulz-Snyder algorithm for recovering a nonnegative function from its autocorrelation is given by the following iteration:

$$\lambda^{(k+1)}(\mathbf{x}) = \lambda^{(k)}(\mathbf{x}) \frac{1}{C(\lambda)} \sum_{\mathbf{y}} \lambda^{(k)}(\mathbf{x}+\mathbf{y}) \frac{S(\mathbf{y})}{R_{\lambda_k}(\mathbf{y})}. \tag{50}$$

Note that if $\lambda_0(\mathbf{x}) = 0$ for some particular $\mathbf{x}$, then $\lambda_k(\mathbf{x}) = 0$ for that $\mathbf{x}$ for all k. This provides a convenient way of incorporating support constraints when they are available. This algorithm possesses some other useful properties such as monotonically decreasing I-divergence and conservation of total intensity of estimates, and its fixed points are (global or local) minimizers of Eq. (46) [93]. Another noteworthy property is that the Schulz-Snyder algorithm operates completely in the spatial domain, instead of alternating between the Fourier and spatial domains as in Fienup's algorithm [33].

5.2.2 An Algorithm for Aliased Autocorrelations

In x-ray crystallography, we can only measure an "aliased" autocorrelation, unlike in astronomical imaging. This is because the measured Fourier magnitudes in x-ray crystallography are undersampled because of the periodicity of molecular structures [77, 116]. The aliased autocorrelations are called Patterson functions [45, 85, 86].

Fortunately, if we replace the "unaliased" autocorrelation with Patterson functions, all the nice properties of Eq. (50) remain, and hence, the Schulz-Snyder algorithm can also be applied to x-ray crystallography (and other applications with periodic structures) with some tweaks (see Chapter 2). Since it is not difficult to prove that the algorithm's properties remain valid, we omit the proofs for conciseness. The arguments for the proofs are similar to those in Schulz and Snyder [93].

The functions of interest in x-ray crystallography are three-dimensional; this chapter presents theory and simulations in two-dimensions for conciseness and ease of visual presentation. All concepts are readily extended to three dimensions. We use different notations for the aliased and unaliased cases to avoid confusing the two cases. Consider a two-dimensional periodic function $g(\mathbf{r})$. Again, we consider a discretization of all functions involved for computational purposes. Since g is periodic, $\mathbf{r}$ extends from negative infinity to positive infinity. For simplicity, however, we assume $\mathbf{r} = (r_1, r_2)$ ranges over a single period: $1 \le r_1 \le d_1$ and $1 \le r_2 \le d_2$, where r_1 and r_2 take on real values, and d_1 and d_2 are real constants, and $\mathbf{d} = (d_1, d_2)$ represents the period of g. Eq. (50) can be adapted to the periodic case as:

$$\rho^{(k+1)}(\mathbf{r}) = \rho^{(k)}(\mathbf{r}) \frac{1}{C(g)} \sum_{\mathbf{u}} \rho^{(k)}((\mathbf{r}+\mathbf{u}) \bmod \mathbf{d}) \frac{P(\mathbf{u})}{P_{\rho^{(k)}}(\mathbf{u})}, \tag{51}$$

where P denotes the measured Patterson function, obtained directly from the diffraction measurements via an inverse Fourier transform of the squared magnitudes of the diffraction data, P_{ρ_k} denotes the Patterson function of the k-th estimate ρ_k, $\mathbf{u} = (u, v)$ denotes coordinates in the Patterson space (which are also assumed to take on values over one period $[\mathbf{0d}]$), and $C(\rho_k)$ is given by

$$\begin{aligned} C(\rho_k) = \sum_{\mathbf{r}} \rho_k(\mathbf{r}) \ &= \ C(g), \ \forall k, \\ \rho \geq 0, & \end{aligned} \tag{52}$$

where $C(\rho_k)^2 = \sum_{\mathbf{u}} P(\mathbf{u})$. The Patterson function of a periodic function g is defined by

$$P(\mathbf{u}) = \sum_{\mathbf{r}} g((\mathbf{r}+\mathbf{u}) \bmod \mathbf{d}) g(\mathbf{r}), \tag{53}$$

which has the same period as g. Note that Eq. (51) still enjoys monotonically decreasing I-divergence.

Even though Eq. (51) preserves all the nice properties of Eq. (50), there still may be some troublesome issues such as nonunique solutions, where there may exist two different electron density maps that produce the same Patterson function, and convergence to local minima that are not global minima, where the iterations may become trapped in "wrong" answers. Eq. (50) also suffers from similar problems as studied in Chapter 4, but these problems may be more serious in Eq. (51).

5.3 Constrained Phase Retrieval Algorithms

When the data are noisy, the algorithms' estimates tend to be rough, as we illustrate in our experiments. To alleviate this roughness, we incorporate additional constraints via penalty methods, particularly Good's roughness [39] and total variation [90] penalties.

When a penalty is incorporated into the objective functions J, our goal becomes finding λ_0 or ρ_0 such that

$$\lambda_0 = \arg\min_{\lambda \geq 0} I(S||R_\lambda) + \alpha\Phi(\lambda), \tag{54}$$

$$\rho_0 = \arg\min_{\rho \geq 0} I(P||P_\rho) + \beta\Phi(\rho), \tag{55}$$

where S and P are the measured unaliased autocorrelation and the measured Patterson function, respectively, α and β are regularization parameters, and the Φs are functions that depend on the penalty type.

For brevity, we describe and discuss our methods in terms of Eq. (54), which involves unaliased autocorrelations. Nonetheless, the methods can be easily applied to the case of aliased autocorrelations as well.

5.3.1 Penalties towards Smoothness

5.3.1.1 Good's Roughness Penalty

Good's roughness was originally proposed for non-parametric probability density estimation [39].

In the continuous spatial domain, it can be defined as [104]

$$\Phi_G(\lambda) = -\int\int \lambda(x_1, x_2)\left(\frac{\partial^2}{\partial x_1^2} + \frac{\partial^2}{\partial x_2^2}\right)\log\lambda(x_1, x_2)dx_1dx_2, \tag{56}$$

where (x_1, x_2) represents a continuous spatial coordinate (making a slight abuse of notation). Discretizing this expression yields

$$\begin{aligned}\Phi_G(\lambda) = -\sum_{x_1}\sum_{x_2}\lambda(x_1, x_2)\{&\log\lambda(x_1+1, x_2) + \log\lambda(x_1-1, x_2),\\ &+\log\lambda(x_1, x_2+1) + \log\lambda(x_1, x_2-1) - 4\log\lambda(x_1, x_2)\}.\end{aligned} \tag{57}$$

O'Sullivan provided an inspiring alternative interpretation of this discretized penalty [83]. He noted that this discretization of Good's roughness can be equivalently expressed in terms of the I-divergences between neighboring pixels:

$$\Phi_O(\lambda) = I(\lambda||S_V\lambda) + I(\lambda||S_V^{-1}\lambda) + I(\lambda||S_H\lambda) + I(\lambda||S_H^{-1}\lambda), \tag{58}$$

where

$$
\begin{aligned}
\left[S_V \lambda\right](x_1, x_2) &= \lambda((x_1 - 1) \bmod N, x_2), \\
\left[S_V^{-1} \lambda\right](x_1, x_2) &= \lambda((x_1 + 1) \bmod N, x_2), \\
\left[S_H \lambda\right](x_1, x_2) &= \lambda(x_1, (x_2 - 1) \bmod M), \\
\left[S_H^{-1} \lambda\right](x_1, x_2) &= \lambda(x_1, (x_2 + 1) \bmod M), \qquad (x_1, x_2) \in \mathcal{X}.
\end{aligned} \tag{59}
$$

From Eq. (58), it can be inferred that if there are large differences between neighboring pixels, then the roughness penalty encourages smoothness by suppressing the differences in the sense of the *I*-divergence, which makes Good's roughness particularly attractive in our overall framework.

5.3.1.2 Total Variation Penalty

Although Good's roughness penalty nicely smoothes contiguous regions in the estimates, it often undesirably smears out edges. Total variation penalties can often provide smoothness while preserving edges [90, 113], since it only weakly penalizes large discontinuities.

Total variation is given, in its continuous form, by

$$
\Phi_{TV}(\lambda) = \int\int |\nabla \lambda(x_1, x_2)|_2 dx_1 dx_2, \tag{60}
$$

where $|\cdot|_2$ denotes the Euclidean norm in $\mathbb{R}^2$, and $\nabla\lambda$ denotes the gradient of λ [46, 59]. Following the suggestions by Combettes and Luo [19, 20], we employ the following dicretization of the total variation penalty:

$$
\begin{aligned}
\Phi_{TV}(\lambda) = &\sum_{x_1=1}^{N-1} \sum_{x_2=1}^{M-1} \sqrt{|\lambda(x_1+1, x_2) - \lambda(x_1, x_2)|^2 + |\lambda(x_1, x_2+1) - \lambda(x_1, x_2)|^2} \\
&+ \sum_{x_1=1}^{N-1} |\lambda(x_1+1, M) - \lambda(x_1, M)| + \sum_{x_2=1}^{M-1} |\lambda(N, x_2+1) - \lambda(N, x_2)|,
\end{aligned} \tag{61}
$$

where the second and third terms on the right-hand side take into account boundary effects [19, p. 1299].

5.3.2 A Relation between EM algorithms and Minimum I-divergence Algorithms

Recall that we aim to find the λ_0 that attains

$$\lambda_0 = \arg\min_{\lambda \geq 0} I(S||R_\lambda) + \alpha\Phi(\lambda). \tag{62}$$

Note the following relations:

$$\begin{aligned}
&\arg\min_{\lambda \geq 0} I(S||R_\lambda) + \alpha\Phi(\lambda) \\
&= \arg\min_{\lambda \geq 0} \sum_{\mathbf{y}} \left\{ S(\mathbf{y}) \log \frac{S(\mathbf{y})}{R_\lambda(\mathbf{y})} - S(\mathbf{y}) + R_\lambda(\mathbf{y}) \right\} + \alpha\Phi(\lambda) \\
&= \arg\min_{\lambda \geq 0} \sum_{\mathbf{y}} \{S(\mathbf{y}) \log S(\mathbf{y}) - S(\mathbf{y})\} - \sum_{\mathbf{y}} \{S(\mathbf{y}) \log R_\lambda(\mathbf{y}) - R_\lambda(\mathbf{y})\} + \alpha\Phi(\lambda) \\
&= \arg\max_{\lambda \geq 0} \sum_{\mathbf{y}} \{S(\mathbf{y}) \log R_\lambda(\mathbf{y}) - R_\lambda(\mathbf{y})\} - \alpha\Phi(\lambda),
\end{aligned} \tag{63}$$

where the last equality is satisfied since $\sum_{\mathbf{y}} [S(\mathbf{y}) \log S(\mathbf{y}) - S(\mathbf{y})]$ does not depend on λ. Note that the last line in Eq. (63) corresponds with maximum penalized-likelihood estimation using a Poisson data model [105, 107]. These important relations suggest that a sequence of λ that can achieve maximum penalized-likelihood can also achieve minimum penalized-I-divergence.

Expectation-Maximization (EM) algorithms are strategic tools for producing a sequence of estimates $\lambda^{(k)}$ that try to maximize the penalized likelihood by maximizing $Q(\lambda^{(new)}; \lambda^{(old)}) - \alpha\Phi(\lambda^{(new)})$ at each iteration, where

$$Q(\lambda^{(new)}; \lambda^{(old)}) = E\left[L_{cd}(\lambda^{(new)})|z, \lambda^{(old)}\right]. \tag{64}$$

Under typical regularity conditions, this can be done by solving

$$D^{10}Q(\lambda^{(new)}; \lambda^{(old)}) - \alpha D\Phi(\lambda^{(new)}) = 0. \tag{65}$$

In the formulas above, D denotes the derivative operator with respect to the parameters involved (*e.g.*, $D^{10}Q(\lambda^{(new)}; \lambda^{(old)})$ denotes the first-order partial derivative of Q with respect to $\lambda^{(new)}$), Q is the expectation of the loglikelihood $L_{cd}(\lambda^{(new)})$ of hypothetical "complete data" given the current estimate of the parameter $\lambda^{(old)}$ and the measured "incomplete data" z. For the complete description of this setting and notation, readers may refer to the work by Green [41] and the work by Dempster *et al.* [24]. The specific complete data formulation appropriate for the Schulz-Snyder algorithm is given in [92].

Therefore, exactly the same sequence $\lambda^{(k)}$ produced by the EM algorithms can be used to minimize the penalized I-divergence, provided that the EM algorithms are designed by assuming the Poisson data model in [92]. Section 5.3.4 exploits this theoretical connection to adapt Green's OSL methods to the penalized I-divergence optimization problem given in Eq. (54).

5.3.3 Optimization Challenge: Coupling

When $\alpha = 0$, Eq. (65) has a closed-form solution; when $\alpha > 0$, for most penalties, Eq. (65) cannot be solved in closed form. In the case of our spatial penalties, Eq. (65) represents a *coupled* set of linear equations. The derivative of O'Sullivan's version of Good's roughness penalty in Eq. (57) is given by

$$\begin{aligned}
\frac{\partial \Phi_G(\lambda)}{\partial \lambda(\mathbf{x})} &= 4\{1 + \log \lambda(x_1, x_2)\} \\
&- \left\{ \log \lambda(x_1 - 1, x_2) + \frac{\lambda(x_1 + 1, x_2)}{\lambda(x_1, x_2)} \right\} \\
&- \left\{ \log \lambda(x_1 + 1, x_2) + \frac{\lambda(x_1 - 1, x_2)}{\lambda(x_1, x_2)} \right\} \\
&- \left\{ \log \lambda(x_1, x_2 - 1) + \frac{\lambda(x_1, x_2 + 1)}{\lambda(x_1, x_2)} \right\} \\
&- \left\{ \log \lambda(x_1, x_2 + 1) + \frac{\lambda(x_1, x_2 - 1)}{\lambda(x_1, x_2)} \right\}. \qquad (66)
\end{aligned}$$

Note that the derivative of Good's roughness penalty involves all the neighboring pixels of $\lambda(x_1, x_2)$. Hence, a closed-form solution of Eq. (65) is intractable.

A similar situation occurs when the TV penalty is applied. Consider the derivative of the TV penalty:

$$\frac{\partial \Phi_{TV}(\lambda)}{\partial \lambda(\mathbf{x})} = \begin{cases} A(\lambda), & 1 \leq x_1 \leq N-1, 1 \leq x_2 \leq M-1 \\ B(\lambda), & x_1 = N, x_2 \neq M \\ C(\lambda), & x_1 \neq N, x_2 = M \\ 2, & \{\lambda(N,M) - \lambda(N-1,M)\}\{\lambda(N,M) - \lambda(N,M-1)\} \geq 0 \\ 0, & \{\lambda(N,M) - \lambda(N-1,M)\}\{\lambda(N,M) - \lambda(N,M-1)\} \leq 0 \end{cases}, \qquad (67)$$

where

$$
\begin{aligned}
A(\lambda) &= \frac{\lambda(x_1,x_2)-\lambda(x_1-1,x_2)}{\sqrt{|\lambda(x_1,x_2)-\lambda(x_1-1,x_2)|^2+|\lambda(x_1-1,x_2+1)-\lambda(x_1-1,x_2)|^2}} \\
&+\frac{2\lambda(x_1,x_2)-\lambda(x_1+1,x_2)-\lambda(x_1,x_2+1)}{\sqrt{|\lambda(x_1+1,x_2)-\lambda(x_1,x_2)|^2+|\lambda(x_1,x_2+1)-\lambda(x_1,x_2)|^2}} \\
&+\frac{\lambda(x_1,x_2)-\lambda(x_1,x_2-1)}{\sqrt{|\lambda(x_1+1,x_2-1)-\lambda(x_1,x_2-1)|^2+|\lambda(x_1,x_2)-\lambda(x_1,x_2-1)|^2}}, \\
B(\lambda) &= \frac{\lambda(N,x_2)-\lambda(N-1,x_2)}{\sqrt{|\lambda(N,x_2)-\lambda(N-1,x_2)|^2+|\lambda(N-1,x_2+1)-\lambda(N-1,x_2)|^2}}, \\
C(\lambda) &= \frac{\lambda(x_1,M)-\lambda(x_1,M-1)}{\sqrt{|\lambda(x_1+1,M-1)-\lambda(x_1,M-1)|^2+|\lambda(x_1,M)-\lambda(x_1,M-1)|^2}}.
\end{aligned}
\tag{68}
$$

The coupling problem becomes even more complicated, and no closed-form solution of Eq. (65) is available.

Various methods such as gradient based methods [56, 75, 104] can be used to maximize the penalized complete-data loglikelihood, hence solving Eq. (65). O'Sullivan suggested a generalized EM algorithm for solving the coupling problems caused by Good's-roughness-type neighborhood structures based on *coloring* ideas [5, 6, 83]. Green's one-step-late (OSL) algorithm is another method, which is straightforward to apply and implement. The connection between minimum penalized-I-divergence algorithms and penalized EM algorithms. This justifies application of the OSL algorithms to our framework.

5.3.4 Green's One-step-late (OSL) Algorithms

Green's OSL algorithms were originally tweaks of EM algorithms designed for maximum penalized-likelihood estimation. Green [41] noted that the relevant objective function in an EM algorithm may be linearized at the current estimate as in gradient methods, and the derivatives of the penalty term at the two consecutive iterations bear small differences if the algorithm converges slowly, as is the case with EM algorithms [24] and other multiplicative algorithms [66]. In an EM formulation, these observations suggest finding a parameter λ that satisfies

$$
D^{10}Q(\lambda^{(new)}|\lambda^{(old)}) - \alpha D\Phi(\lambda^{(old)}) = 0. \tag{69}
$$

Notice the only difference between Eqs. (65) and (69) is that $\lambda^{(old)}$ is used in Eq. (69) instead of $\lambda^{(new)}$. An appealing property of the OSL algorithm is that Eqs. (69) and (65) have the same fixed points.

Green's OSL algorithms have empirically shown monotonically increasing penalized-likelihood functions of incomplete data and show faster convergence rate per iteration compared with the associated unconstrained EM algorithms. However, we emphasize that the faster convergence speed is due to the penalty, rather than the attributes of OSL algorithm.

Since the sequence of estimates generated by OSL algorithms can attain maximum penalized-likelihood, they can also achieve minimum penalized-I-divergence as we discussed earlier. Thus, the next section derives the algorithms for minimizing the penalized I-divergence objective function given in Eq. (54) by exploiting the OSL idea.

5.3.5 Constrained Phase Retrieval Algorithms

The unconstrained minimum I-divergence algorithms can be interpreted as deterministic versions [84] of the EM algorithm associated with the Poisson data model. Consequently, the unconstrained minimum I-divergence algorithms inherit the corresponding slow convergence rates of EM algorithms. This encourages us to adapt Green's OSL algorithm to our minimum I-divergence methods to perform the Constrained I-divergence minimization given in Eq. (54).

Application of Green's OSL idea yields the algorithms

$$\lambda^{(k+1)}(\mathbf{x}) = \frac{\lambda^{(k)}(\mathbf{x})}{2C(\lambda) + \alpha D\Phi(\lambda)\big|_{\lambda=\lambda^{(k)}}} 2\sum_{\mathbf{y}} \lambda(\mathbf{x}+\mathbf{y}) \frac{S(\mathbf{y})}{R_{\lambda}^{(k)}(\mathbf{y})}. \tag{70}$$

Following a similar idea, the algorithm for a Patterson function can be obtained:

$$\rho^{(k+1)}(\mathbf{r}) = \frac{\rho^{(k)}(\mathbf{r})}{2C(\rho) + \alpha D\Phi(\rho)\big|_{\rho=\rho^{(k)}}} 2\sum_{\mathbf{u}} \rho((\mathbf{r}+\mathbf{u}) \bmod \mathbf{d}) \frac{P(\mathbf{u})}{P_{\rho}^{(k)}(\mathbf{u})}. \tag{71}$$

We omit the details for brevity; they are a straightforward adaptation of the ideas in [41] to the discussion in Section 5.3.2.

5.4 Numerical Experiments

5.4.1 Experimental Settings

5.4.1.1 Initial Estimates

The Schulz-Snyder phase retrieval algorithm in Eq. (50) and our modification in Eq. (51) of the algorithm for x-ray crystallography are both subject to a serious challange, namely convergence to local minima (see Chapters 2 and 4). In addition, it is generally difficult to know whether an estimate is a local or a global minimum. Hence, in our experiments, we initialize the algorithms with a known truth added to a small constant ϵ. This allows us to focus on the effects of noise and regularization without becoming confused by issues involving local minima (which haunt all practical phase retrieval algorithms). Finding methods for avoiding local minima is very challenging; it is an active research problem.

Note that we do not assume that the exact image support is known. For example, suppose a circle contained in a 32×32 rectangle is the true image, and the region other than the circle in the rectangle is filled with zeros. Then, we may initialize the algorithms with a 32×32 constant rectangle plus the true image, which is assumed to be known for purposes of our study. By doing this, we can isolate the effects of noise propagation through the algorithms from the problem of convergence to local minima, since we start from a place that is probably near a global minimum.

The size of the initial estimate should be carefully chosen. Note that the summation term on the right-hand side of Eq. (50) contains not only the autocorrelations but also the cross-correlation of an estimate with an autocorrelation. When we compute the autocorrelation of an image using FFT-based convolutions, the size of the resulting image should be set to at least twice that of the original image to avoid having some part of the autocorrelation "wrap around" undesirably. The same logic can be used for the cross-correlation. Since the estimate is cross-correlated with the autocorrelation (whose size is twice that of the estimate) of itself, the resulting image size should be set to at least 3 times that of the estimate. Therefore, we should begin with an initial image whose size is $3N \times 3N$, where $2N \times 2N$ is the size of the measured autocorrelation. Note there is a lot of zero padding in the initial estimate. The nonzero part of the initial estimate that we explained how to construct in the last paragraph is placed at the center of the whole initial estimate.

When Patterson functions are involved, both autocorrelations and cross-correlations are computed using circular convolutions, therefore initial estimates for the algorithm in Eq. (51) have the same size as that of the known truth.

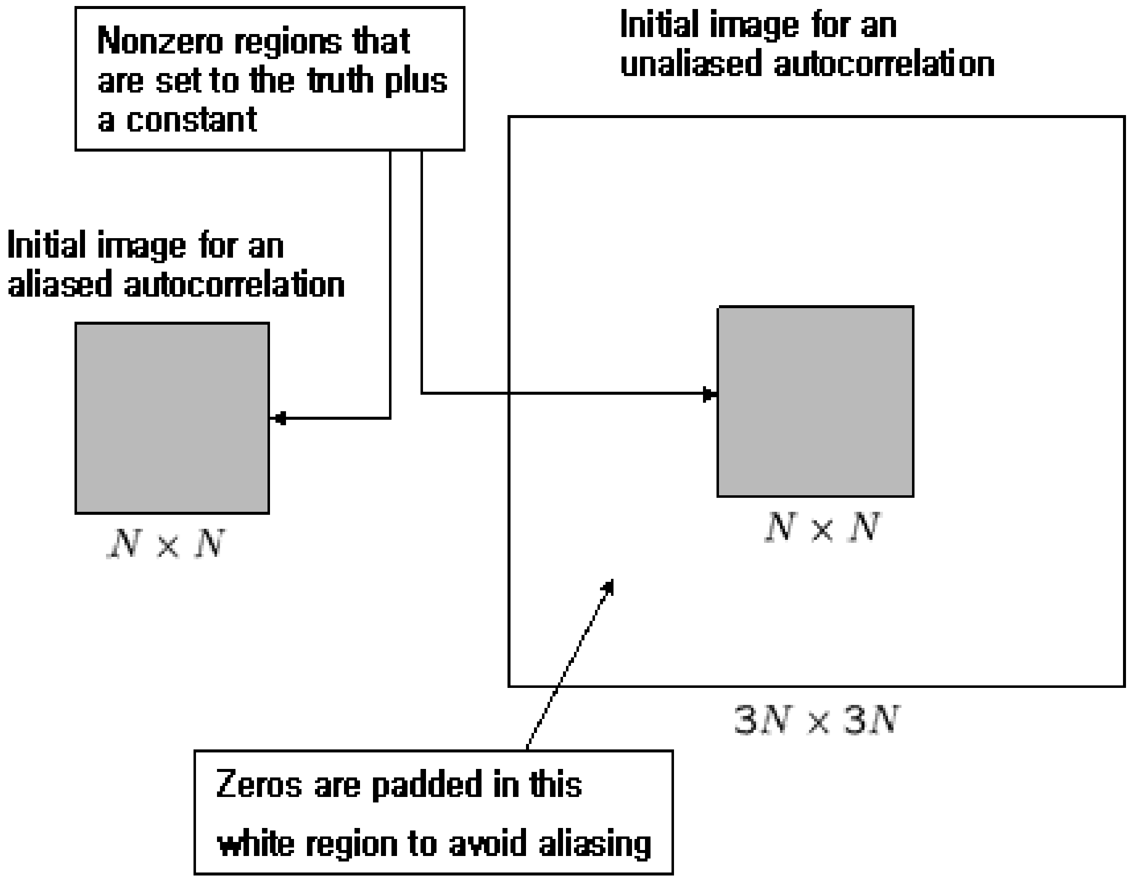


Figure 54: Example of initial estimates: The initial estimate on the left is used for the algorithm in Eq. (51), and that on the right is used for the algorithm in Eq. (50).

Figure 54 shows an example of initial estimates for both aliased and unaliased autocorrelations. Again note that the exact true support is not known for both cases. Also note that zeros are padded to avoid the "wraparound" problem of autocorrelations or cross-correlations in the "unaliased" case.

5.4.1.2 Noisy Data Realization

When measurements are recorded as photon counts, as in astronomical imaging or x-ray crystallography, relevant noise may be modeled by Poisson statistics. However, we may observe diverse noise artifacts since noise corrupts measurements in different ways dependent upon the application.

Poisson Noise on Autocorrelations: Fig. 55 shows the procedure through which we generate noisy autocorrelations. A series of 2×2 arrays connected by solid arrow lines represents the procedure for generating an aliased, noisy autocorrelation; the combination of two 2×2 arrays and three 3×3 arrays connected by dotted, arrow lines represents the procedure for generating an unaliased, noisy autocorrelation.

From probability theory, the SNR for a Poisson random variable is related to the mean μ of the random variable provided that the mean is large: $SNR \propto \sqrt{\mu}$ [29]. Therefore, we may control noise levels by changing the image intensities, which act as means for Poisson random variables. Since we are interested in the change of noise artifacts with respect to the change of noise levels, we first change the image intensity level by scaling the known truth f by a constant c: $f_c = cf$. (Here, we use a generic f instead of λ or ρ, since the exposition may apply to both cases.)

Noiseless autocorrelations P_{f_c} and R_{f_c} are produced for both aliased and unaliased cases using f_c according to Eqs. (47) and (53). Note that the sizes for two autocorrelations are different. Then, noisy autocorrelations are generated by

$$\begin{aligned} P_{f_c}^{noisy}(\mathbf{u}) &\sim Poisson(P_{f_c}(\mathbf{u})), \\ R_{f_c}^{noisy}(\mathbf{y}) &\sim Poisson(R_{f_c}(\mathbf{y})). \end{aligned} \tag{72}$$

That is, each pixel of P_{f_c} and R_{f_c} is the mean of the corresponding pixels of $P_{f_c}^{noisy}$ and $R_{f_c}^{noisy}$, respectively. Note that the algorithms assume autocorrelations are symmetric. However, since each pixel is associated with different realization, $P_{f_c}^{noisy}$ and $R_{f_c}^{noisy}$ are not symmetric. Therefore, we enforce symmetries for the autocorrelations as follows:

$$\begin{aligned} Sym(P_{f_c}^{noisy})(\mathbf{u}) = Sym(P_{f_c}^{noisy})(-\mathbf{u}) &= \frac{P_{f_c}^{noisy}(\mathbf{u}) + P_{f_c}^{noisy}(-\mathbf{u})}{2} \\ Sym(R_{f_c}^{noisy})(\mathbf{y}) = Sym(R_{f_c}^{noisy})(-\mathbf{y}) &= \frac{R_{f_c}^{noisy}(\mathbf{y}) + R_{f_c}^{noisy}(-\mathbf{y})}{2}. \end{aligned} \tag{73}$$

Data in astronomical imaging through extreme turbulence may be generated as $R_{f_c}^{noisy}$. We do not know of any physical mechanism that generates data as in $P_{f_c}^{noisy}$; we include it to facilitate an "apples-to-apples" comparison between the unaliased and aliased cases. Data in the aliased case is typically physically generated as described next.

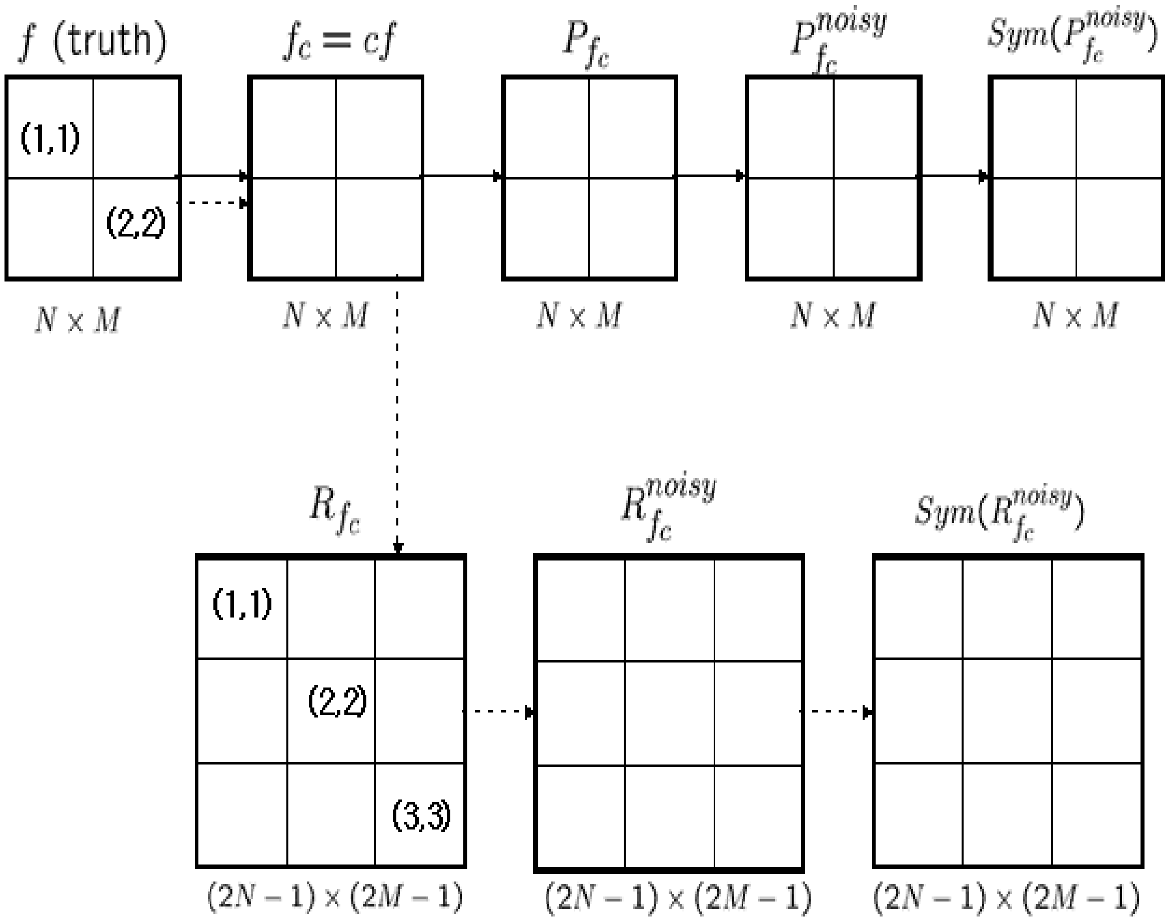


Figure 55: Procedure for realizing noisy autocorrelations: an unaliased noisy autocorrelation is generated by the procedure indicated by the solid arrows; an aliased noisy autocorrelation is generated by the procedure indicated by the dotted arrows.

Poisson Noise on Squared Fourier Magnitudes: Figure 56 shows the procedure for generating an aliased, noisy autocorrelation. However, in this case, Poisson noise is added to squared Fourier magnitudes. Let I_{f_c} denote the Fourier magnitudes squared: $I_{f_c} = |\mathcal{F}\{f_c\}|^2$, where $\mathcal{F}\{\cdot\}$ denotes the Fourier transform operator and $f_c = cf$ as before. Then, noisy Fourier magnitudes are produced by

$$I_{f_c}^{noisy}(\omega) \sim Poisson(I_{f_c}(\omega)), \tag{74}$$

where ω represents 2-D frequency-domain coordinates. An aliased, noisy autocorrelation is then obtained by

$$P_{f_c}^{noisy} = \mathcal{F}^{-1}\{I_{f_c}\}, \tag{75}$$

where $\mathcal{F}^{-1}\{\cdot\}$ denotes the inverse Fourier transform operator. Note that the autocorrelation $P_{f_c}^{noisy}$ in this case obeys the correct symmetry $P_{f_c}^{noisy}(\mathbf{u}) = P_{f_c}^{noisy}(-\mathbf{u})$.

5.4.1.3 Error Metrics

For qualification of the deterioration of estimates by noise, we study changes via various error metrics:

$$\begin{aligned}
L_1(f,\lambda) &= \sum_{\mathbf{x}} |f(\mathbf{x}) - \lambda(\mathbf{x})|, \\
L_2(f,\lambda) &= \sum_{\mathbf{x}} |f(\mathbf{x}) - \lambda(\mathbf{x})|^2, \\
L_\infty(f,\lambda) &= \max_{\mathbf{x}} |f(\mathbf{x}) - \lambda(\mathbf{x})|, \\
I(f||\lambda) &= \sum_{\mathbf{x}} \left\{ f(\mathbf{x}) \log \frac{f(\mathbf{x})}{\lambda(\mathbf{x})} - f(\mathbf{x}) + \lambda(\mathbf{x}) \right\},
\end{aligned} \tag{76}$$

where f denotes the truth, and λ denotes an estimate. The same error metrics are used for g and ρ, where g denotes the truth and ρ denotes an estimate in the Patterson case. Note that here we use the I-divergence as a discrepancy between the truth and an estimate rather than their autocorrelations.

When the truth is scaled to control the noise level in the autocorrelations, the error metric is also scaled. Hence, we use tweaks of the error metrics for fair comparisons.

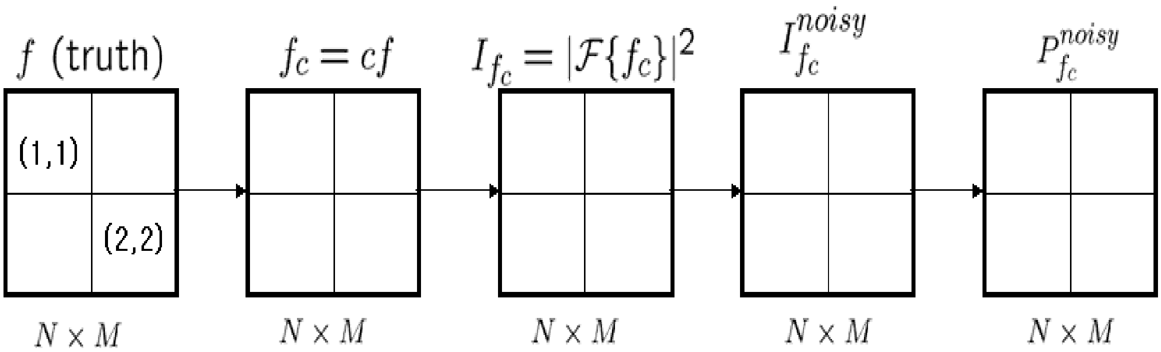


Figure 56: Alternative procedure for realizing noisy aliased autocorrelations, where Poisson noise is added to Fourier magnitudes, and the noisy, aliased autocorrelation is obtained by taking the inverse Fourier transform to the noisy magnitudes.

Observe that

$$
\begin{aligned}
L_1(f_c, \lambda_c) &= \sum_{\mathbf{x}} |cf(\mathbf{x}) - c\lambda(\mathbf{x})|, \\
L_2(f_c, \lambda_c) &= \sum_{\mathbf{x}} |cf(\mathbf{x}) - c\lambda(\mathbf{x})|^2, \\
L_\infty(f_c, \lambda_c) &= \max_{\mathbf{x}} |cf(\mathbf{x}) - c\lambda(\mathbf{x})|, \\
I(f_c||\lambda_c) &= \sum_{\mathbf{x}} \left\{ cf(\mathbf{x}) \log \frac{cf(\mathbf{x})}{c\lambda(\mathbf{x})} - cf(\mathbf{x}) + c\lambda(\mathbf{x}) \right\}, \qquad (77)
\end{aligned}
$$

where f_c denotes a scaled truth, and λ_c denotes an estimate of f_c. Therefore, we can argue that

$$
\begin{aligned}
L_1(f_c, \lambda_c) &= cL_1(f, \lambda), \\
L_2(f_c, \lambda_c) &= c^2 L_2(f, \lambda), \\
L_\infty(f_c, \lambda_c) &= cL_\infty(f, \lambda), \\
I(f_c||\lambda_c) &= cI(f||\lambda). \qquad (78)
\end{aligned}
$$

Thus, when we compare $L_1(f, \lambda)$ and $L_1(f_c, \lambda_c)$, we divide $L_1(f_c, \lambda_c)$ by c. For the other metrics, the same reasoning and method is used.

5.4.2 Unconstrained Estimates

5.4.2.1 Poisson Noise on Autocorrelations

Unconstrained Reconstructions from Unaliased Autocorrelations: Figure 57 shows the truth (a hand image) and its aliased and unaliased autocorrelations. The colormaps for the autocorrelations are modified to best show details and are provided on the right-hand sides of the autocorrelations. Figure 58 shows selected estimates produced by Eq. (50) from unaliased autocorrelations for various c values. Recall that the SNR becomes lower as c becomes smaller. Among the estimates in Fig. 58, the estimate for $c = 0.01$ is associated with the lowest SNR. Observe that estimate become rougher as the SNR becomes lower. For $c = 0.01$, it is difficult to recognize the hand in the estimate.

Different noise realizations would result in different estimates. To obtain the "average" behavior of the algorithm in noise, we perform 10 Monte Carlo experiments. Even though the estimates in Fig. 58 are selected from the 10 experiments, the other estimates for the

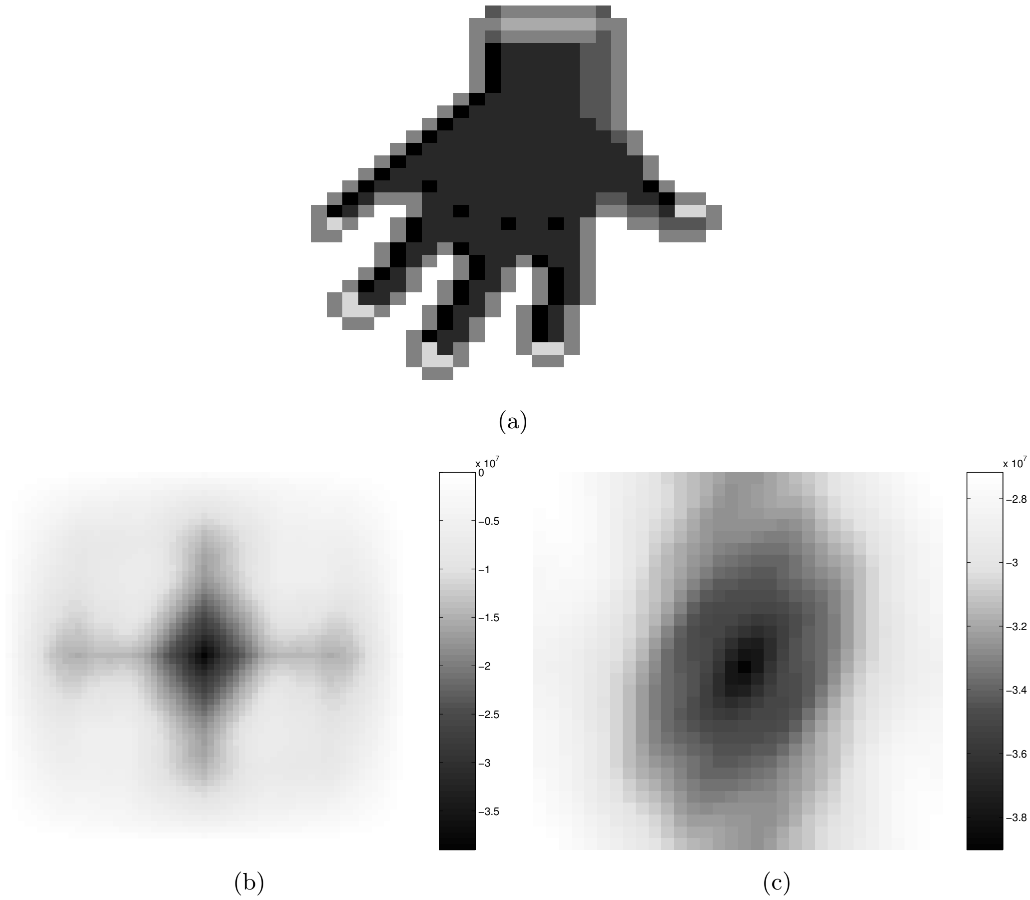


Figure 57: (a) Truth image. (b) Unaliased autocorrelation of the truth in Fig. 57(a). (c) Aliased autocorrelation (or Patterson function) of the truth in Fig. 57(a). The colormaps of autocorrelations are modified to best show details; the colormaps are given on the right of the autocorrelation images.

same c have similar quality. Figure 59 shows the mean images of the estimates from the 10 Monte Carlo experiments for the c values in Fig. 58. As expected, estimate quality is better in the mean images. Compare the estimates in Figs. 58(f) and 59(f): The hand in the mean image for $c = 0.01$ is quite distinguishable. Figure 60 shows the pixelwise variance images of the estimates from the 10 Monte Carlo experiments. Overall, variances of estimates are higher on the background than on the hand, but as the SNR becomes lower, the variances on the background and on the hand become similar.

Unconstrained Reconstructions from Aliased Autocorrelations: Figure 61 shows selected estimates produced by Eq. (51) from aliased autocorrelations for the same c values

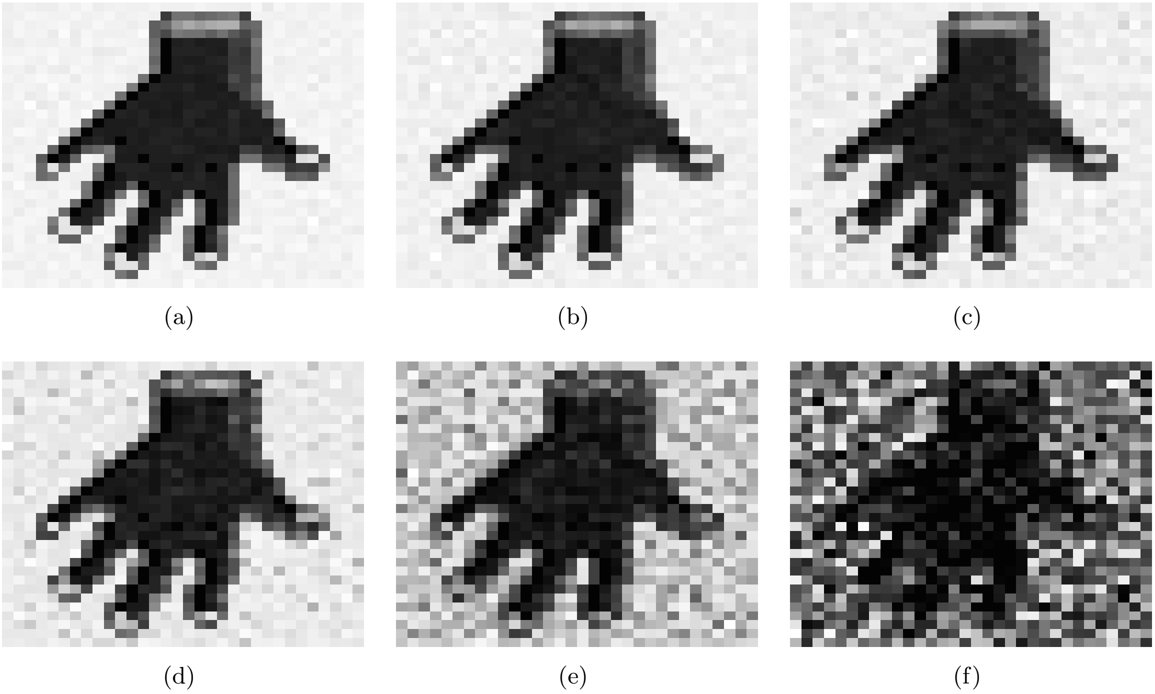


Figure 58: Selected unconstrained estimates at the 50000-*th* iteration produced by Eq. (50) from unaliased autocorrelations when (a) $c = 0.26$, (b) $c = 0.21$, (c) $c = 0.16$, (d) $c = 0.11$, (e) $c = 0.06$, and (f) $c = 0.01$.

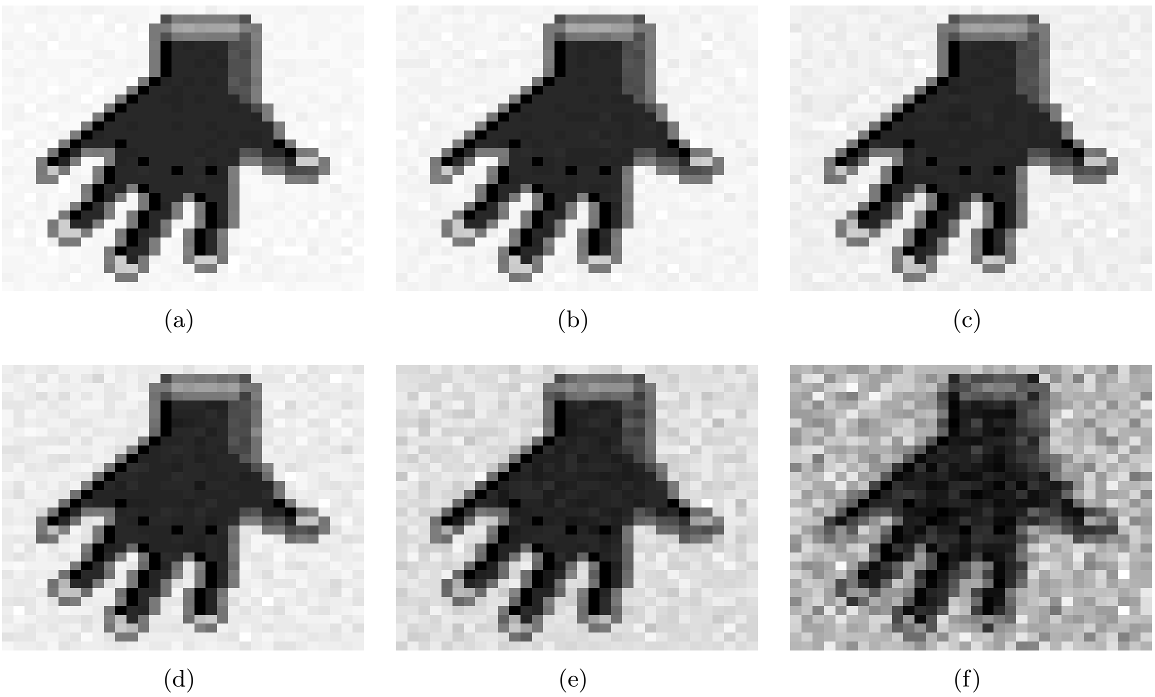
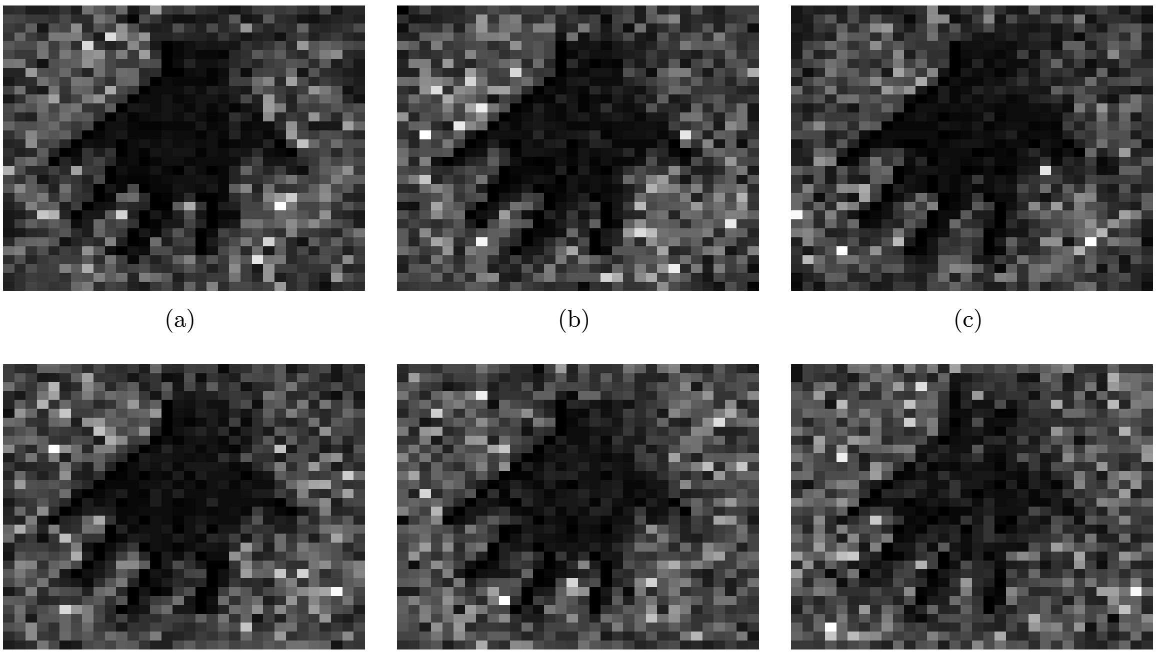


Figure 59: Mean images of unconstrained estimates at the 50000-*th* iteration of 10 Monte Carlo experiments performed with Eq. (50) when (a) $c = 0.26$, (b) $c = 0.21$, (c) $c = 0.16$, (d) $c = 0.11$, (e) $c = 0.06$, and (f) $c = 0.01$. The measured autocorrelations are not aliased.



(a) (b) (c)

(d) (e) (f)

Figure 60: Variance images of unconstrained estimates at the 50000-*th* iteration of 10 Monte Carlo experiments performed with Eq. (50) when (a) $c = 0.26$, (b) $c = 0.21$, (c) $c = 0.16$, (d) $c = 0.11$, (e) $c = 0.06$, and (f) $c = 0.01$. The measured autocorrelations are not aliased.

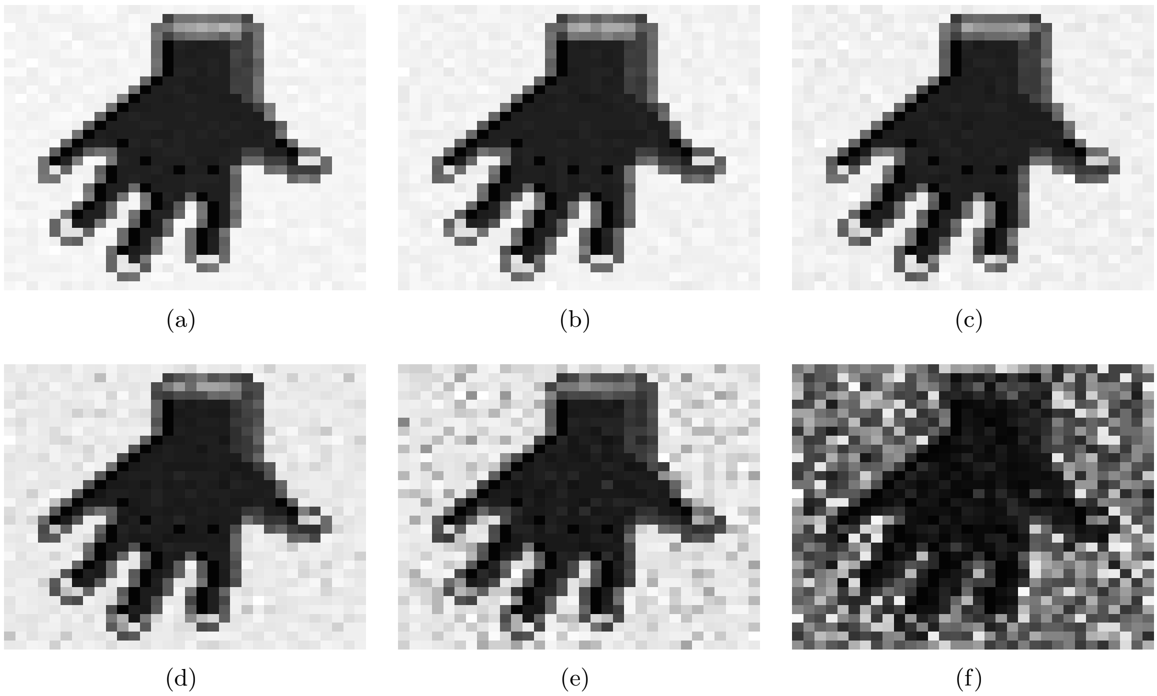
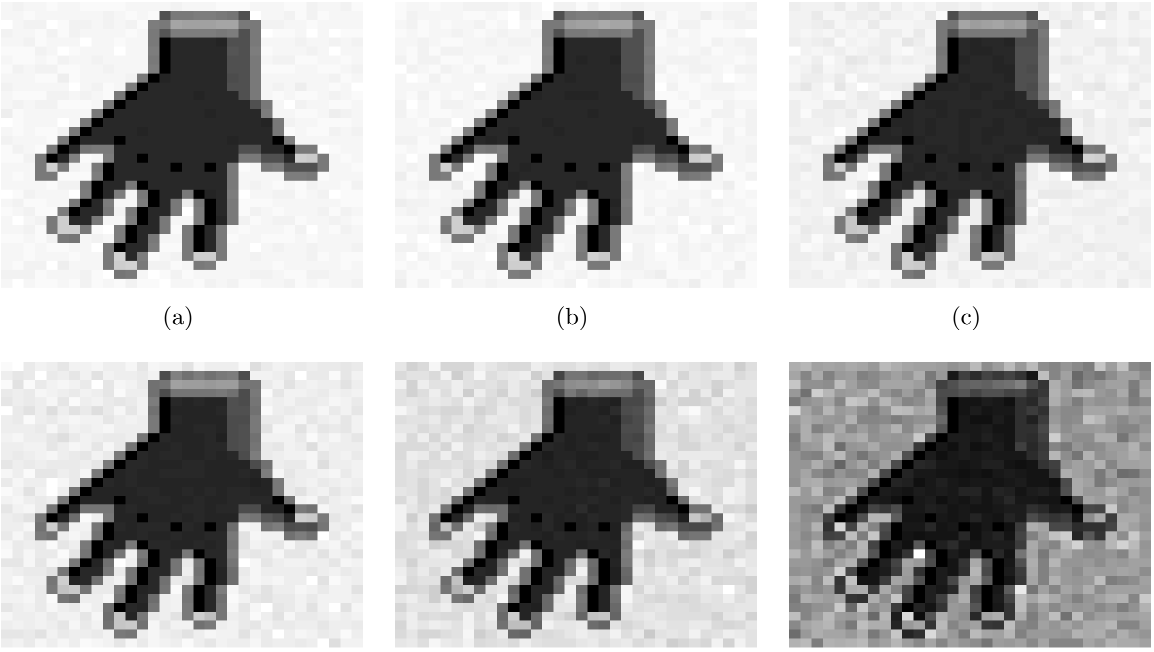


Figure 61: Selected unconstrained estimates at the 50000-*th* iteration produced by Eq. (51) from aliased autocorrelations when (a) $c = 0.26$, (b) $c = 0.21$, (c) $c = 0.16$, (d) $c = 0.11$, (e) $c = 0.06$, and (f) $c = 0.01$.

in Fig. 58. Noticeably, the estimates from aliased autocorrelations suffer less from Poisson noise than those from unaliased autocorrelations. Compare the estimates in Figs. 58 and 61 and observe that the hand in Fig. 61(f) is more distinguishable than that in Fig. 58(f). Also, the background in Fig. 61(e) is less rough than the background in Fig. 58(e). Figures 62 and 63 show the mean and variance images of 10 Monte Carlo runs associated with Fig. 61 for the c values in Fig. 61. Again, as expected, the mean images look better than the estimates in Fig. 61. Note that the mean image for $c = 0.01$ shows much improved smoothness on the background. The variance images show behavior similar to the variances for the case of unaliased autocorrelations.

Error Metric Comparison: Because of the randomness of noise, it is not so obvious that estimates from aliased autocorrelations suffer less from noise than those from unaliased autocorrelations. This may be seen via comparison of error metrics. Figure 64 shows various error metrics that we discussed in Section 5.4.1.3. Each subplot shows the values of an error



(a) (b) (c)

(d) (e) (f)

Figure 62: Mean images of unconstrained estimates at the 50000-*th* iteration of 10 Monte Carlo experiments performed with Eq. (51) when (a) $c = 0.26$, (b) $c = 0.21$, (c) $c = 0.16$, (d) $c = 0.11$, (e) $c = 0.06$, and (f) $c = 0.01$. The measured autocorrelations are aliased.

metric as c changes. Since Monte Carlo runs are involved, the error metric's behavior is represented by two lines: The upper line represents the mean of the 10 Monte Carlo runs plus the standard deviation of the 10 runs, and the lower line represents the mean minus the standard deviation. Now, it is clear that the estimates from aliased autocorrelations are less degraded by noise than the estimates from unaliased autocorrelations in the sense of the four error metrics.

It is important to note that noise "suddenly" destroys estimate quality, as opposed to having a "gradual" impact. Note that the error-metric values suddenly shoot up for c between 0.21 and 0.01.

5.4.2.2 Poisson Noise on Squared Fourier Magnitudes

Figure 65 shows selected estimates produced by the unconstrained algorithm in Eq. (51) from aliased autocorrelations. Here, Poisson noise is generated with means given by the squared Fourier magnitudes of the truth. The aliased autocorrelations are obtained by

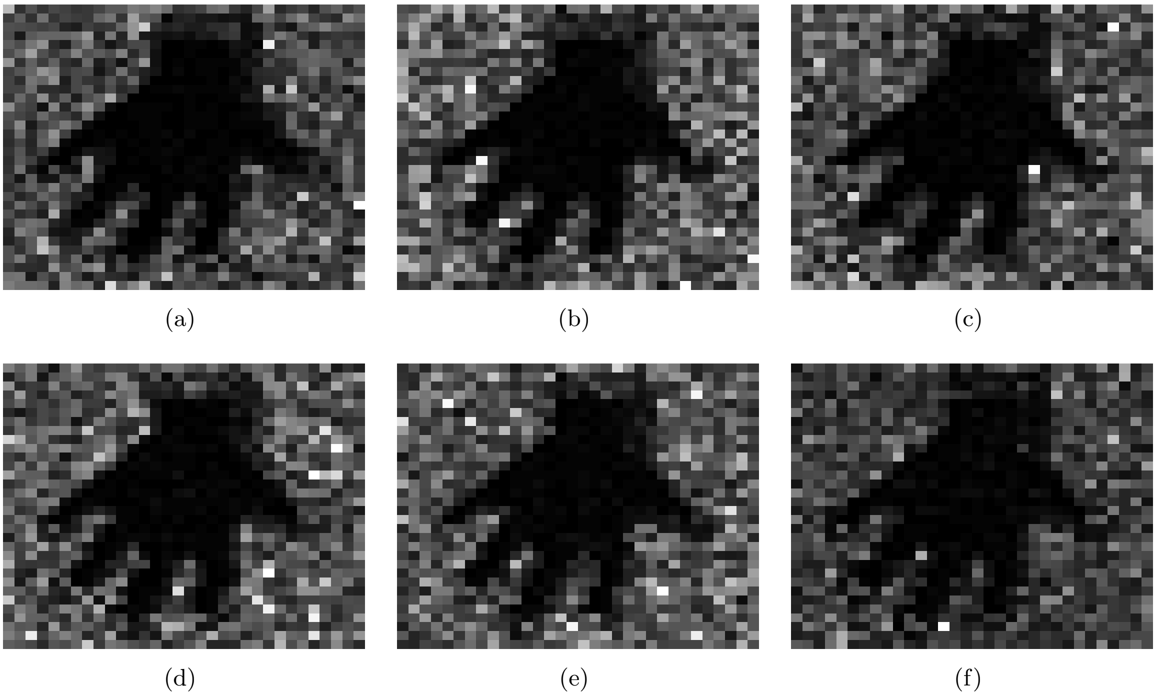


Figure 63: Variance images of unconstrained estimates at the 50000-*th* iteration of 10 Monte Carlo experiments performed with Eq. (51) when (a) $c = 0.26$, (b) $c = 0.21$, (c) $c = 0.16$, (d) $c = 0.11$, (e) $c = 0.06$, and (f) $c = 0.01$. The measured autocorrelations are aliased.

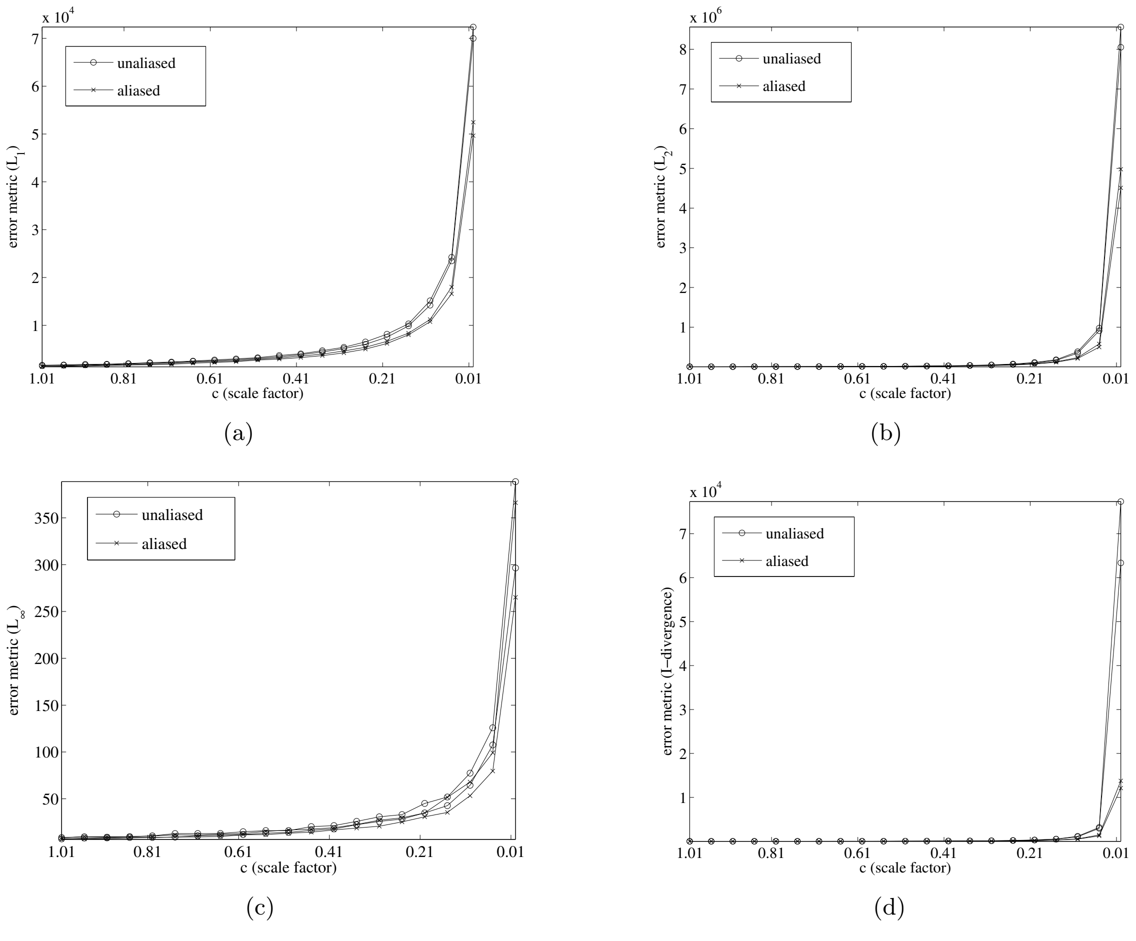


Figure 64: Various error metrics when the autocorrelations are subject to Poisson noise: (a) L_1, (b) L_2, (c) L_∞, and (d) I-divergence.

taking the inverse Fourier transform of the noisy squared Fourier magnitudes. The Fourier magnitudes are assumed to be "undersampled" [77], which results in aliased autocorrelations. Because the values of Fourier magnitudes are large, the c values for the estimates in Fig. 65 are much smaller than those used in Figs. 58 and 61. We can clearly observe roughness in the estimates, especially when c is low, which corresponds to a low SNR.

Comparing c values in this section and the previous section reveals important information: For obtaining a similar SNR level, much lower photon counts would be necessary when noise is placed on the Fourier magnitudes squared compared with when noise is placed on the autocorrelations directly (compare the estimates in Figs. 61(f) and 65(f)).

Figures 66 and 67 show the mean and variance images of 10 Monte Carlo experiments associated with Fig. 65. The mean images again show improved image quality, as expected. The variance images also show similar behavior to those in Figs. 60 and 63.

The error metrics are illustrated in Fig. 68. As in the cases when noise is added to autocorrelations directly, noise suddenly and rapidly degrades estimate quality in the sense of the four error metrics once the noise reaches a critical level.

Unconstrained Reconstructions from Highly Noisy Fourier Data: When SNRs become severely low, interesting noise artifacts that look like sinusoidal patterns occur. Figure 69 shows some selected estimates from low SNR autocorrelations. Since the noise corrupts the Fourier information, noise dominating certain measurements may destroy some frequency components. Dependent upon the particular noise realization, there may be several frequency components destroyed by noise; this results in various artifacts in the autocorrelations as in the second row of Fig. 69. Note the noise artifacts in Fig. 69 look like several types of sinusoidal patterns.

5.4.3 Constrained Estimates

5.4.3.1 Poisson Noise on Autocorrelations

Constrained Estimates from Unaliased Autocorrelations: Figure 70 shows estimates produced by Eq. (70) when Good's roughness is incorporated for a relatively high SNR ($c = 0.06$). When $\alpha = 0.5$, the roughness on the hand seen when the estimate is

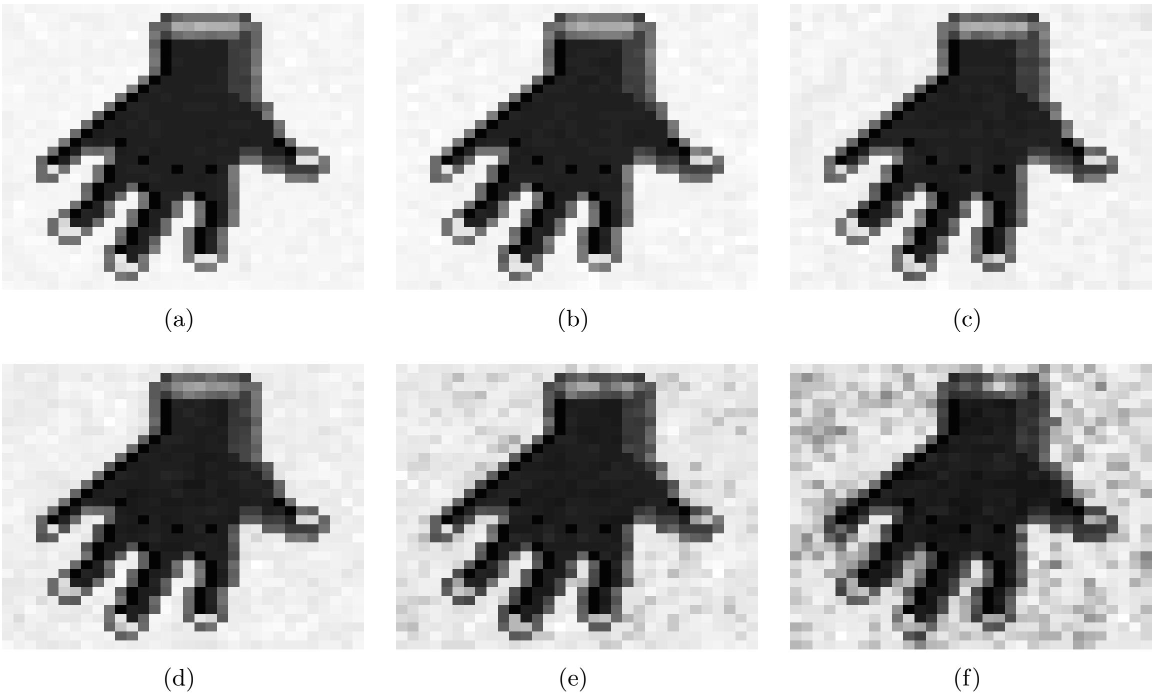


Figure 65: Selected unconstrained estimates at the 50000-*th* iteration produced by Eq. (51) from aliased autocorrelations when (a) $c = 0.001535$, (b) $c = 0.0012875$, (c) $c = 0.00104$, (d) $c = 0.0007925$, (e) $c = 0.000545$, and (f) $c = 0.0002975$.

unconstrained is much alleviated, while the background in the estimate still remains a little rough. Higher α values provide more smoothness to the background, but the hand becomes blurred as α becomes higher. When $\alpha = 5.0$, the border of the hand is too smeared out.

When the SNR is low ($c = 0.01$), noise leads to estimates that are too messy to be distinguishable; Good's roughness cannot help much. Figure 71 shows estimates produced by Eq. (70), incorporating Good's roughness. The estimate produced for $\alpha = 0.5$ starts to show a smooth enough hand, which is more recognizable than the unconstrained estimate in Fig. 58(f). The estimate with $\alpha = 1.0$ achieves nice smoothness both on the hand region and the background. Higher α values overly smooth the estimate; the estimate produced with $\alpha = 5.0$ is not even recognizable as a hand.

Noticeably, the TV penalty induces considerably different smooth textures compared with Good's roughness. Figures 72 and 73 show estimates produced by Eq. (70) when the TV penalty is applied. Observe how flat the estimates are in both the hand and the background regions for α higher than 0.5. Another important property of the TV penalty

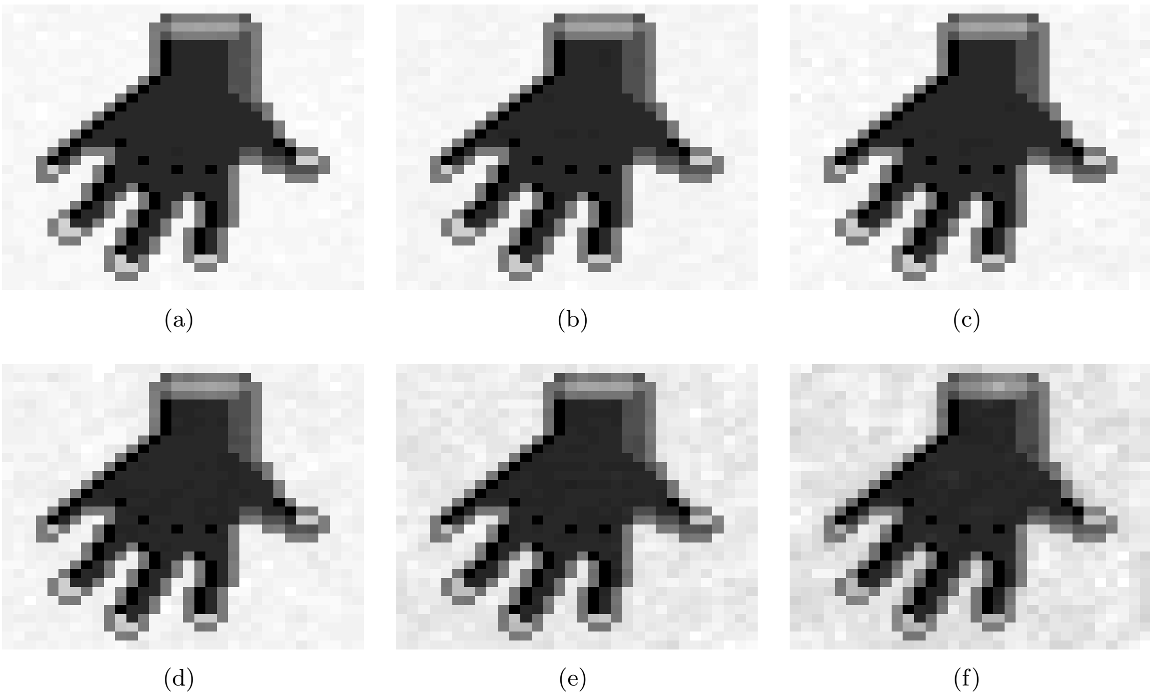


Figure 66: Mean images of unconstrained estimates at the 50000-*th* iteration of 10 Monte Carlo experiments performed with Eq. (51) when (a) c = 0.001535, (b) c = 0.0012875, (c) c = 0.00104, (d) c = 0.0007925, (e) c = 0.000545, and (f) c = 0.0002975. Poisson noise is placed on Fourier magnitudes that are undersampled, resulting in noisy, aliased autocorrelations.

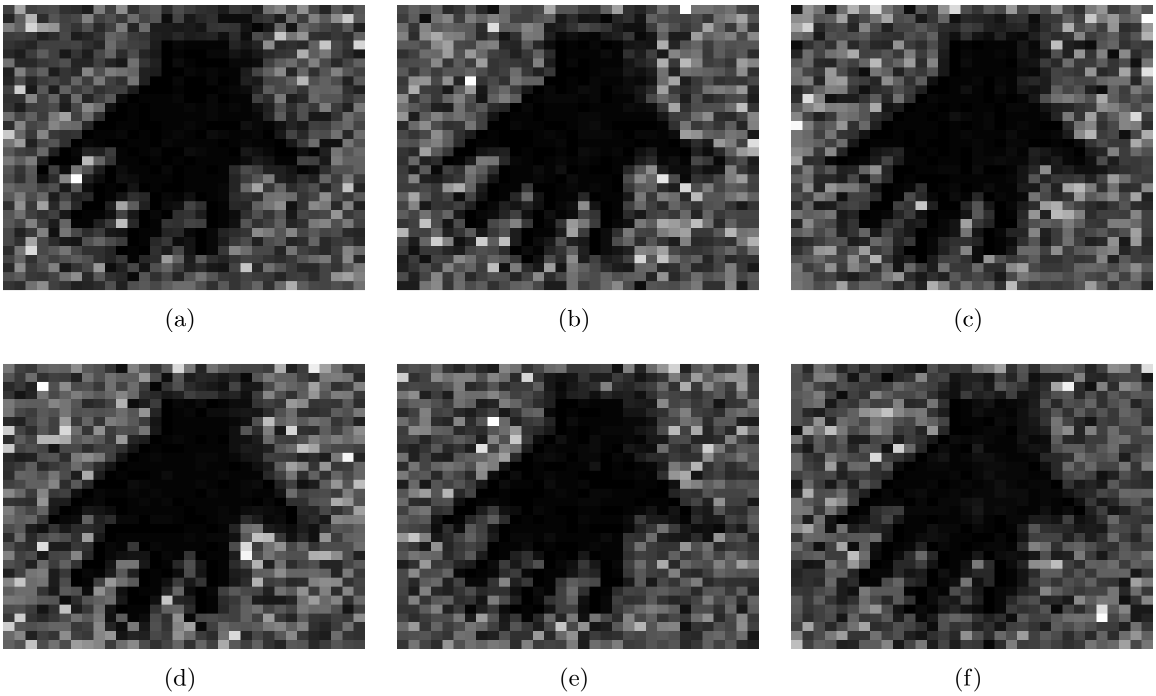


Figure 67: Variance images of unconstrained estimates at the 50000-th iteration of 10 Monte Carlo experiments performed with Eq. (51) when (a) $c = 0.001535$, (b) $c = 0.0012875$, (c) $c = 0.00104$, (d) $c = 0.0007925$, (e) $c = 0.000545$, and (f) $c = 0.0002975$. Poisson noise is placed on squared Fourier magnitudes that are undersampled, resulting in noisy, aliased autocorrelations.

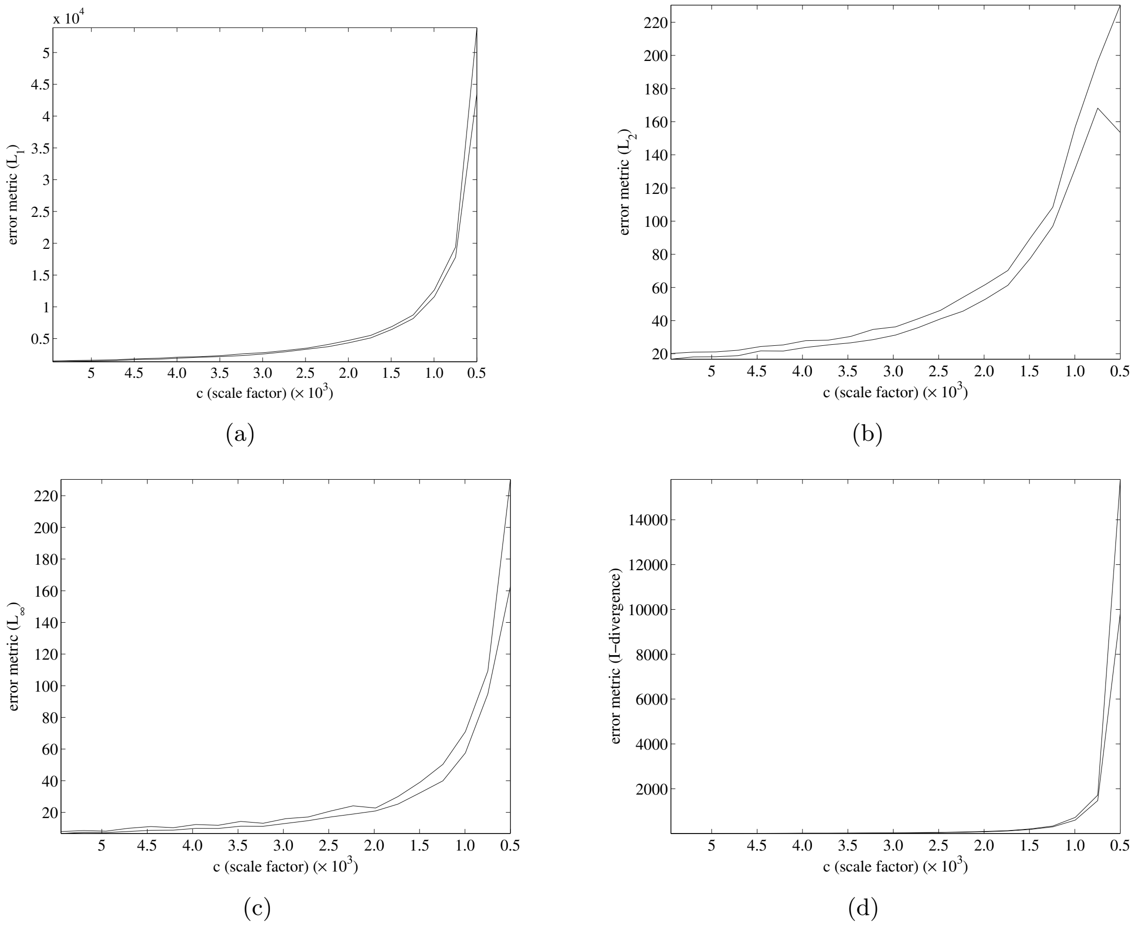


Figure 68: Various error metrics when Poisson noise is placed on squared Fourier magnitudes: (a) L_1, (b) L_2, (c) L_∞, and (d) I-divergence. The occasional jumpiness of the curve (as near the right side of Fig. 68(b)) is due to the limited number of Monte Carlo runs. We did not perform more runs since the overall trends are already quite clear.

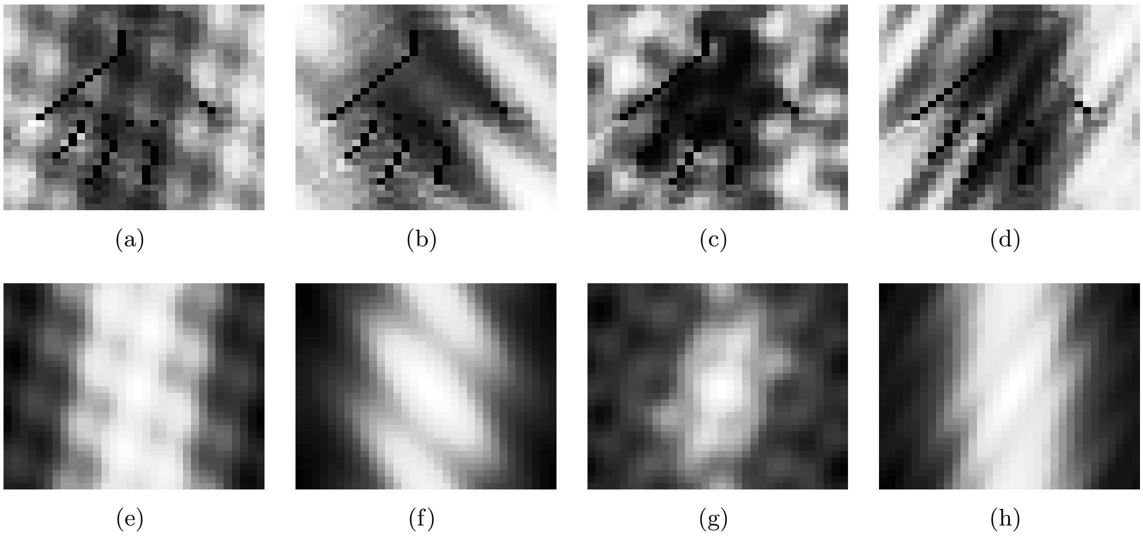
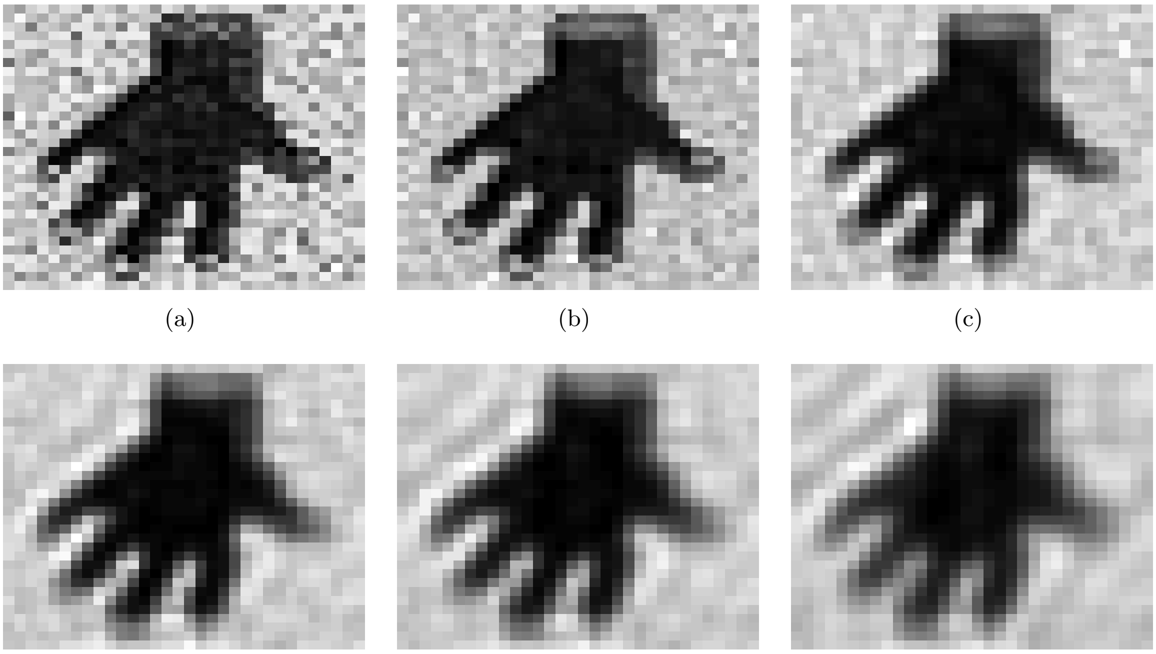


Figure 69: Interesting unconstrained estimates at the 50000-*th* iteration produced by Eq. (51) from aliased autocorrelations with low SNRs when (a) $c = 0.000035$, (b) $c = 0.00004$, (c) $c = 0.000045$, (d) $c = 0.00005$. Poisson noise is placed on squared Fourier magnitudes. The autocorrelations of the estimates in Figs. 69(a), 69(b), 69(c), and 69(d) are shown in Figs. 69(e), 69(f), 69(g), and 69(h), respectively.

is that it preserves edges of estimates. Note the edges are quite clear even for $\alpha = 5.0$ (compare Fig. 72(f) with Fig. 70(f)). However, if the noise level is high ($c = 0.01$), then the TV penalty cannot seem to locate the correct edges for α higher than 2.0. As with Good's roughness, the estimates with the TV penalty for $\alpha = 0.5$ and 1.0 are much improved over the unconstrained estimates and become recognizable.

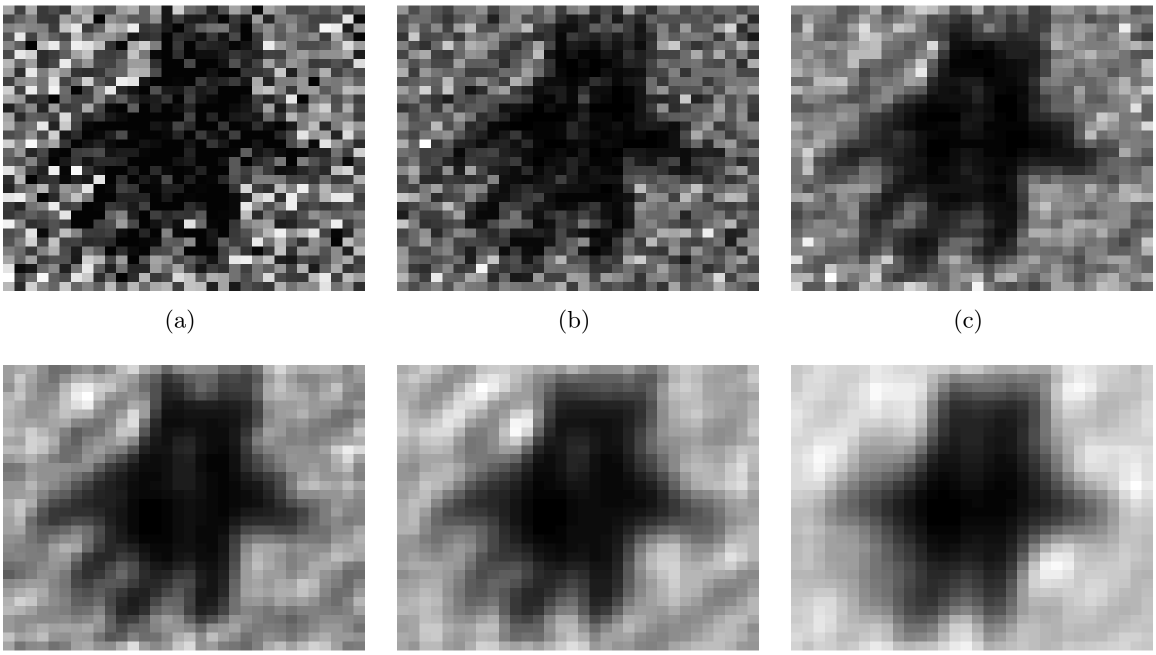
Constrained Estimates from Aliased Autocorrelations: Figures 74 and 75 show estimates produced by Eq. (71) with Good's roughness penalty for $c = 0.06$ and 0.01, respectively, when Poisson noise is placed on aliased autocorrelations. Similarly to the estimates from the unaliased autocorrelations, the penalty leads to smooth estimates for some α. On the other hand, the algorithm in Eq. (71) is more sensitive to the operation of the penalty. Note that the estimate in Fig. 74(f) is a lot more blurred than the estimate in Fig. 70(f), although the noise levels are similar, and the same regularization parameters are applied. Since the estimates produced by Eq. (71) are so sensitively driven by the penalty effects, it could be difficult to find an appropriate α that can provide enough smoothing without smearing out the features of estimate. Observe the estimates in Fig. 75. Even



(a) (b) (c)

(d) (e) (f)

Figure 70: Estimates produced by Eq. (70) incorporating Good's roughness penalty given unaliased autocorrelations when $c = 0.06$ (high SNR), and (a) unconstrained, (b) $\alpha = 0.1$, (c) $\alpha = 0.5$, (d) $\alpha = 1.0$, (e) $\alpha = 2.0$, and (f) $\alpha = 5.0$.



(a) (b) (c)

(d) (e) (f)

Figure 71: Estimates produced by Eq. (70) incorporating Good's roughness penalty given unaliased autocorrelations when $c = 0.01$ (low SNR), and (a) unconstrained, (b) $\alpha = 0.1$, (c) $\alpha = 0.5$, (d) $\alpha = 1.0$, (e) $\alpha = 2.0$, and (f) $\alpha = 5.0$.

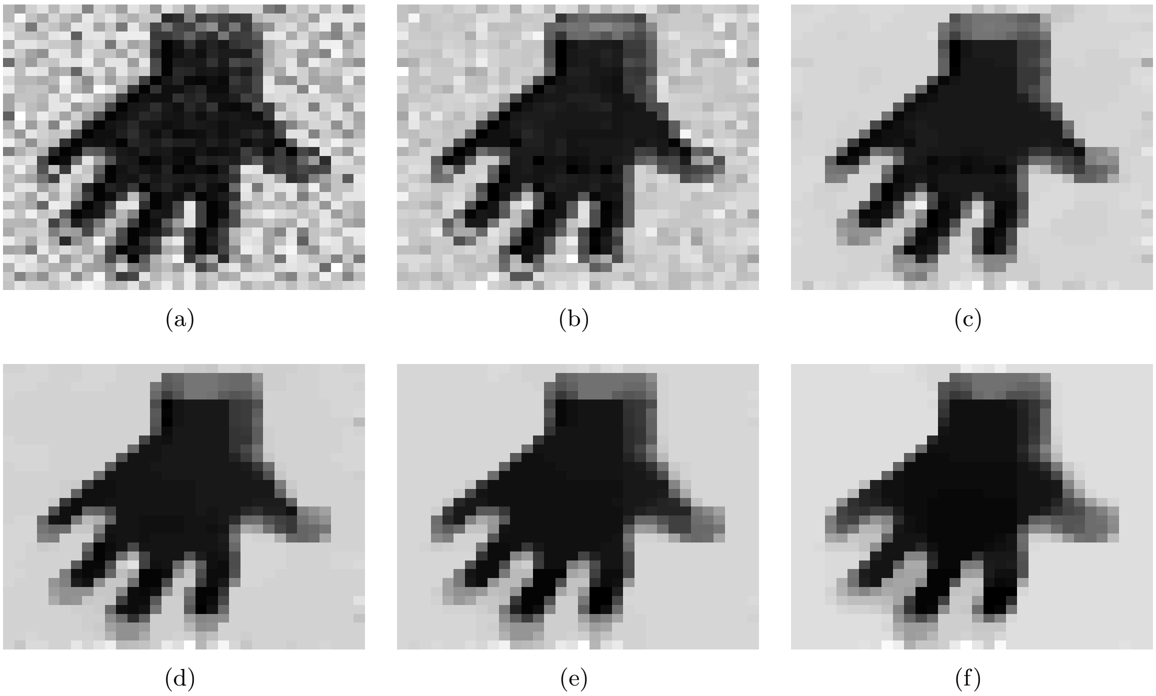


Figure 72: Estimates produced by Eq. (70) incorporating TV penalty given unaliased autocorrelations when $c = 0.06$ (high SNR), and (a) unconstrained, (b) $\alpha = 0.1$, (c) $\alpha = 0.5$, (d) $\alpha = 1.0$, (e) $\alpha = 2.0$, and (f) $\alpha = 5.0$.

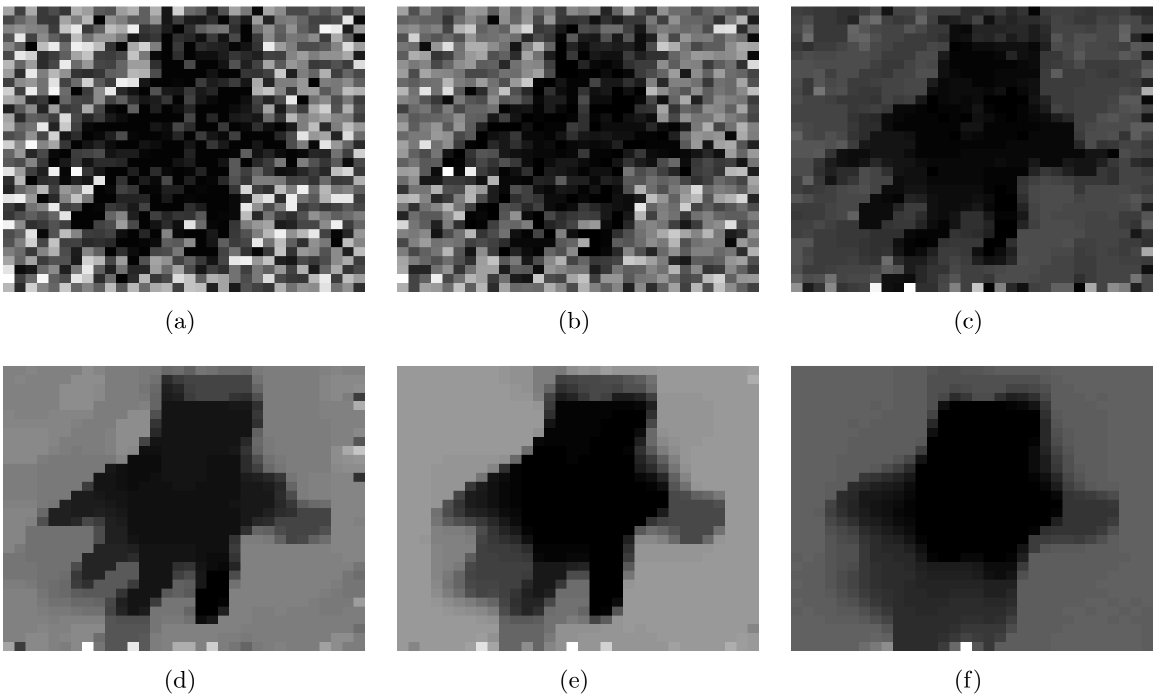


Figure 73: Estimates produced by Eq. (70) incorporating TV penalty given unaliased autocorrelations when $c = 0.01$ (low SNR), and (a) unconstrained, (b) $\alpha = 0.1$, (c) $\alpha = 0.5$, (d) $\alpha = 1.0$, (e) $\alpha = 2.0$, and (f) $\alpha = 5.0$.

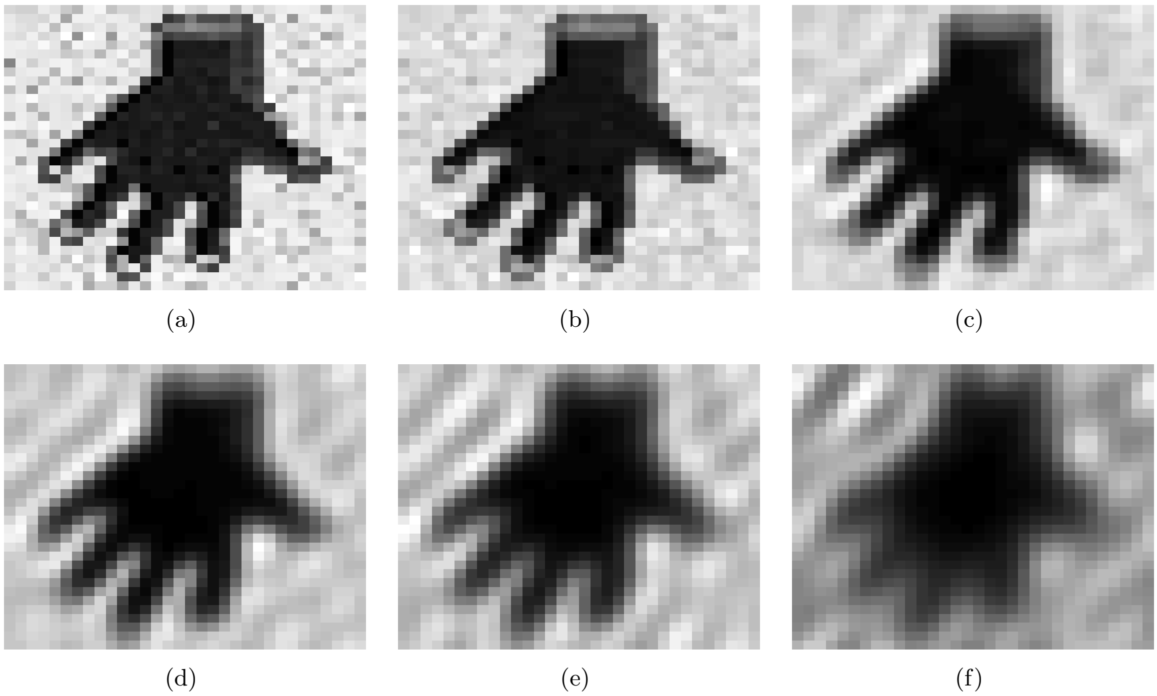


Figure 74: Estimates produced by Eq. (71) incorporating Good's roughness penalty given aliased autocorrelations when $c = 0.06$ (high SNR), and (a) unconstrained, (b) $\alpha = 0.1$, (c) $\alpha = 0.5$, (d) $\alpha = 1.0$, (e) $\alpha = 2.0$, and (f) $\alpha = 5.0$.

though α is low (see the $\alpha = 0.5$ case), the hand becomes quite blurred and starts to lose much of the information about the hand's shape. Obviously, higher α than 0.5 smashes most of the features in the estimates.

Figures 76 and 77 show estimates produced with the TV penalty from aliased autocorrelations. When the SNR is high, the TV penalty suppresses the background roughness due to noise better than Good's roughness; the background becomes almost entirely smooth, and most of the features are well reconstructed (see Fig. 76(c)). A similar sensitivity of the algorithm to the Good's roughness is observed with the TV penalty. When the SNR is low, the algorithm has difficulty in suppressing noise while preserving the features of estimates such as edges, since a slight increase in the regularization parameter could turn the estimates into unrecognizable blobs, as seen in Fig. 77.

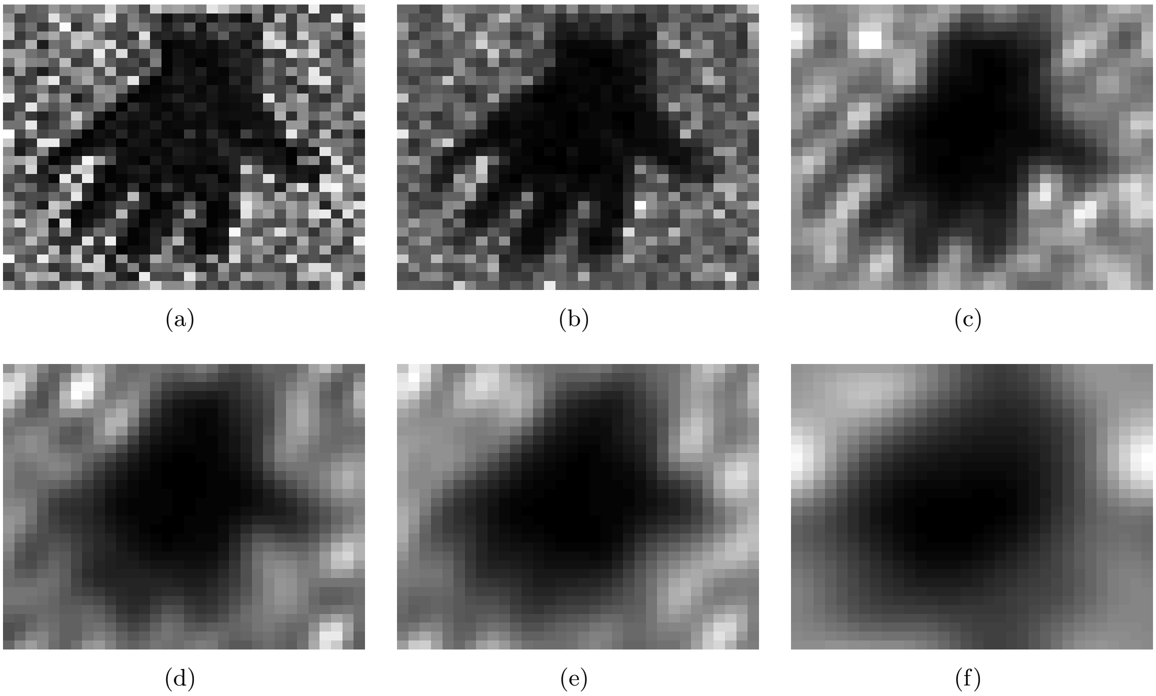


Figure 75: Estimates produced by Eq. (71) incorporating Good's roughness penalty given aliased autocorrelations when $c = 0.01$ (low SNR), and (a) unconstrained, (b) $\alpha = 0.1$, (c) $\alpha = 0.5$, (d) $\alpha = 1.0$, (e) $\alpha = 2.0$, and (f) $\alpha = 5.0$.

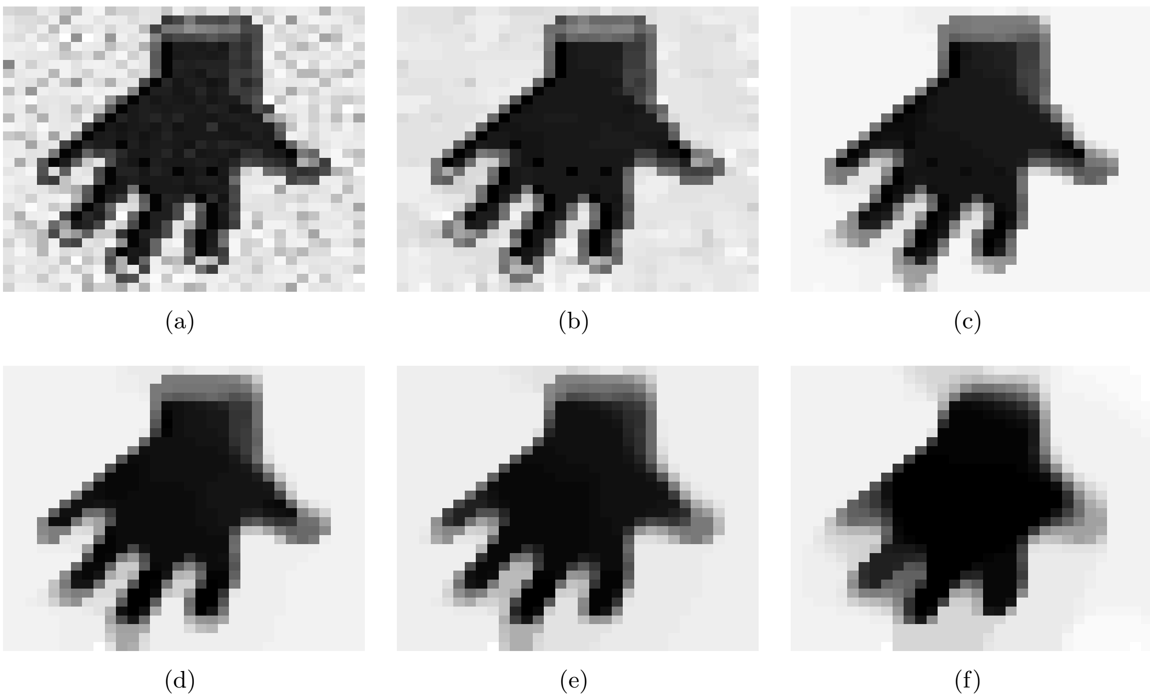


Figure 76: Estimates produced by Eq. (71) incorporating TV penalty given aliased autocorrelations when $c = 0.06$ (high SNR), and (a) unconstrained, (b) $\alpha = 0.1$, (c) $\alpha = 0.5$, (d) $\alpha = 1.0$, (e) $\alpha = 2.0$, and (f) $\alpha = 5.0$.

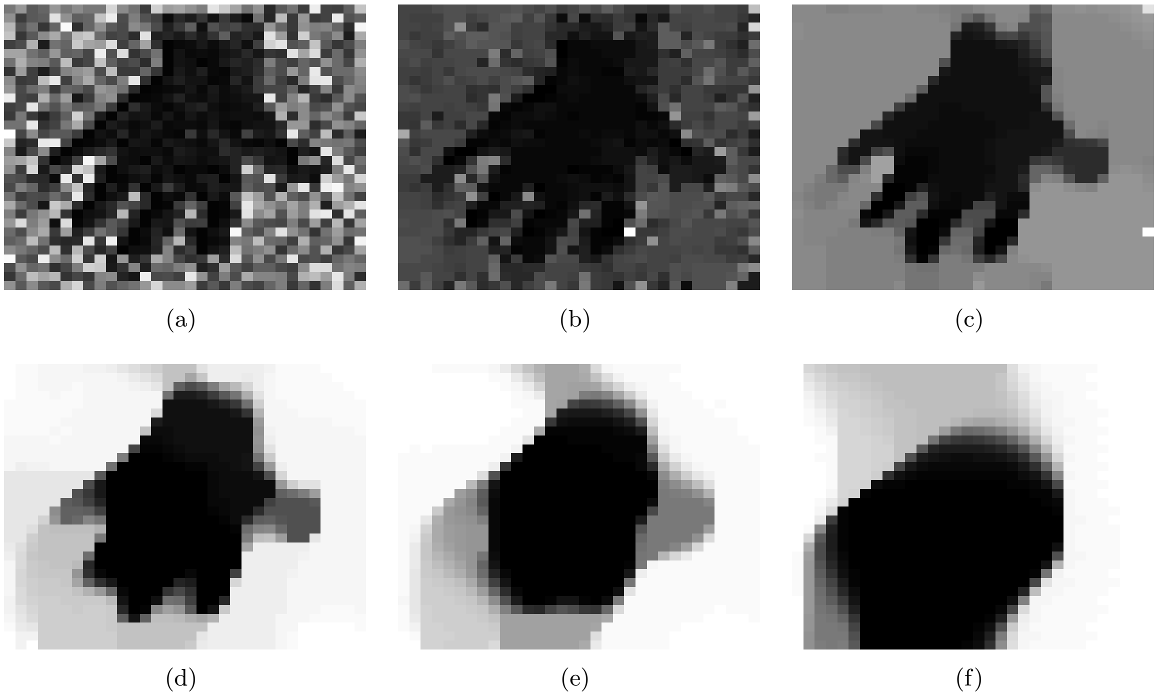


Figure 77: Estimates produced by Eq. (71) incorporating TV penalty given aliased autocorrelations when $c = 0.01$ (low SNR), and (a) unconstrained, (b) $\alpha = 0.1$, (c) $\alpha = 0.5$, (d) $\alpha = 1.0$, (e) $\alpha = 2.0$, and (f) $\alpha = 5.0$.

5.4.3.2 Poisson Noise on Squared Fourier Magnitudes:

Figures 78 through 81 show estimates produced by Eq. (71), incorporating Good's roughness and the TV penalties, in the case where Poisson noise is placed on the squared Fourier magnitudes. The estimates show behavior that is quite similar to the case of aliased autocorrelations in which noise is manifest directly in the spatial domain.

When the SNR is "destructively" low as in Fig. 69, the penalty is helpless no matter what type is used. The unconstrained estimates in Fig. 69 only preserve the edges with large values. A slight amount of smoothing from a penalty entirely blurs out all information in the estimates, and no useful features are observable in the constrained estimates. For brevity, we omit the results of these experiments.

5.5 Conclusions

We studied the effect of noise on phase retrieval via minimization of Csiszár's *I*-divergence in three scenarios of measurements corrupted by Poisson noise. One common artifact in all

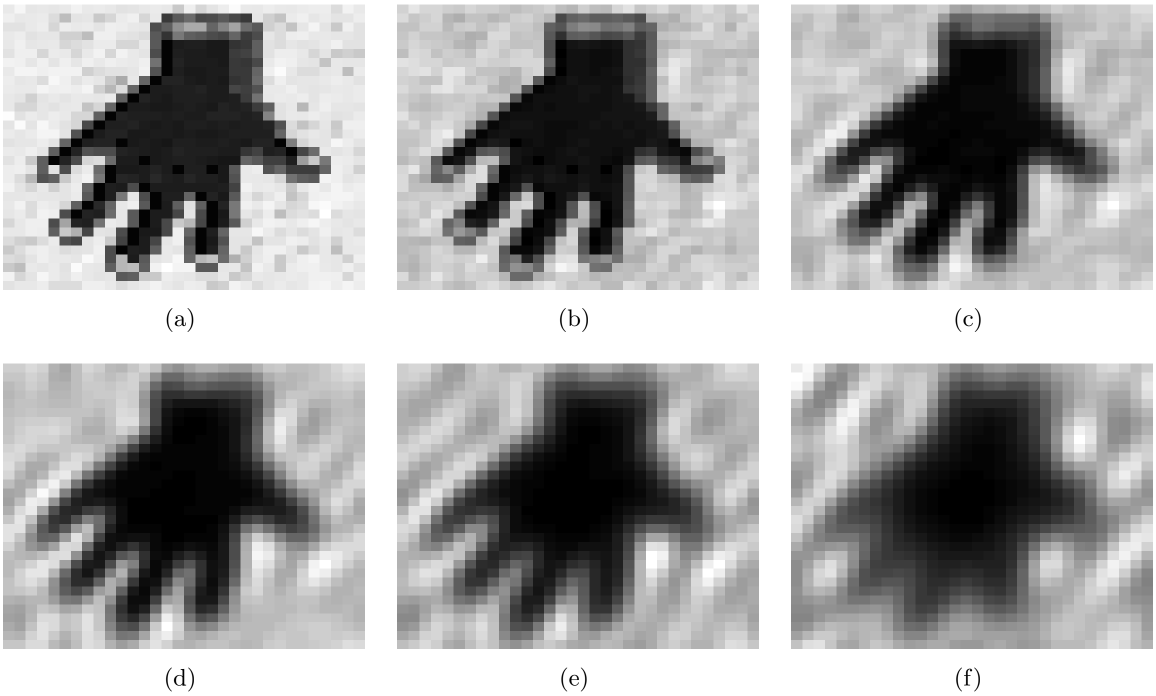


Figure 78: Estimates produced by Eq. (71) incorporating Good's roughness penalty given aliased autocorrelations formed from noisy squared Fourier magnitudes when $c = 0.000545$ (high SNR), and (a) unconstrained, (b) $\alpha = 0.001$, (c) $\alpha = 0.005$, (d) $\alpha = 0.01$, (e) $\alpha = 0.02$, and (f) $\alpha = 0.05$.

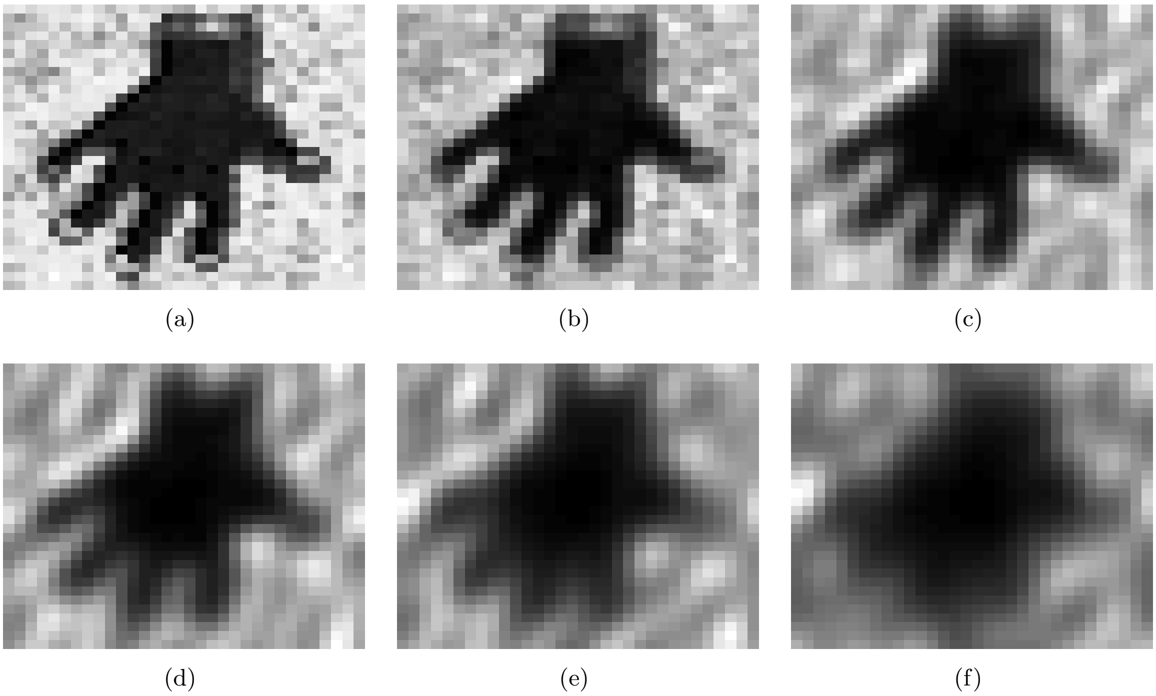
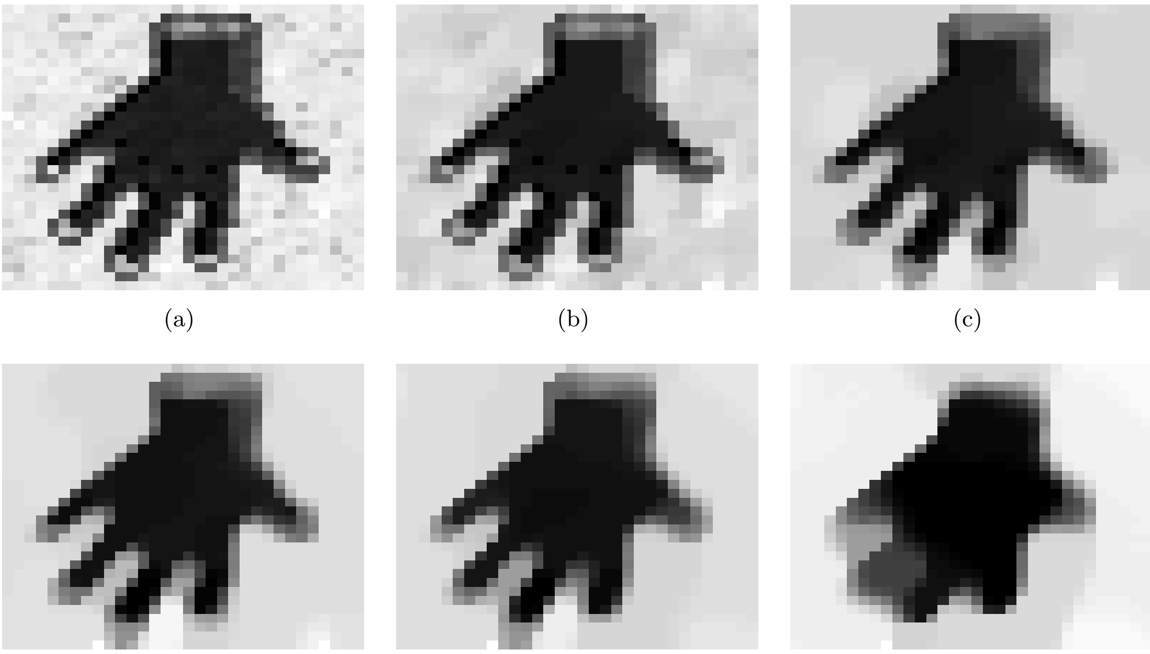


Figure 79: Estimates produced by Eq. (71) incorporating Good's roughness penalty given aliased autocorrelations formed from noisy squared Fourier magnitudes when $c = 0.0002975$ (low SNR), and (a) unconstrained, (b) $\alpha = 0.001$, (c) $\alpha = 0.005$, (d) $\alpha = 0.01$, (e) $\alpha = 0.02$, and (f) $\alpha = 0.05$.



(a) (b) (c)

(d) (e) (f)

Figure 80: Estimates produced by Eq. (71) incorporating TV penalty given aliased autocorrelations formed from noisy squared Fourier magnitudes when $c = 0.000545$ (high SNR), and (a) unconstrained, (b) $\alpha = 0.001$, (c) $\alpha = 0.005$, (d) $\alpha = 0.01$, (e) $\alpha = 0.02$, and (f) $\alpha = 0.05$.

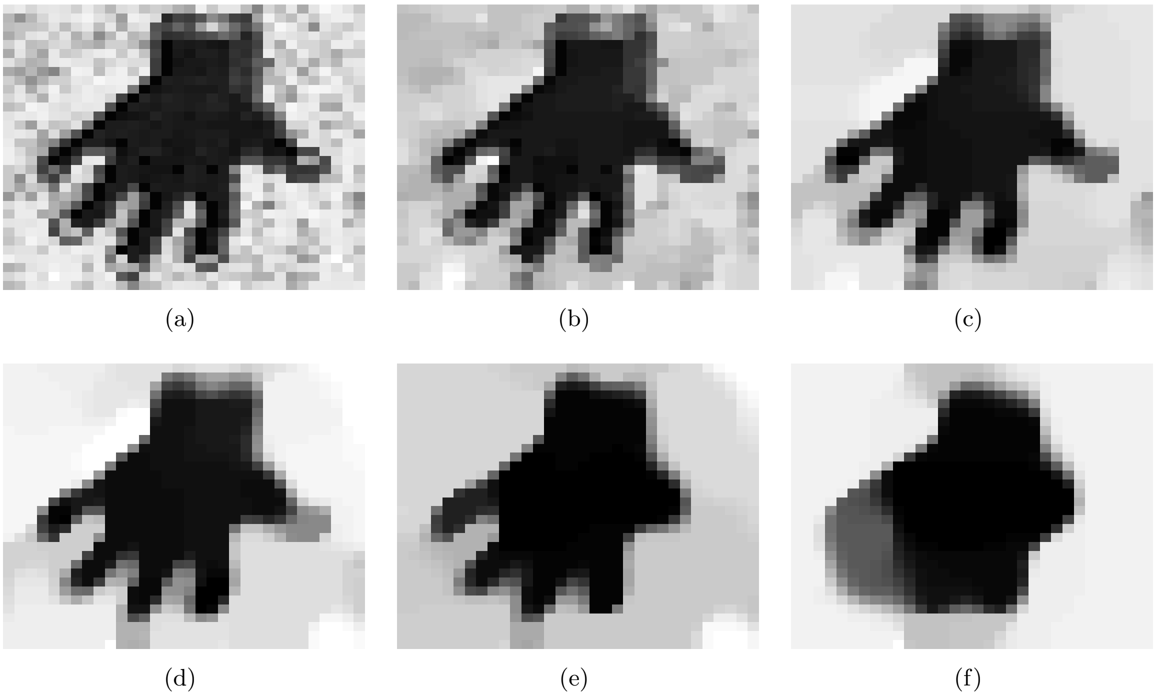


Figure 81 panels: (a), (b), (c), (d), (e), (f)

Figure 81: Estimates produced by Eq. (71) incorporating TV penalty given aliased autocorrelations formed from noisy squared Fourier magnitudes when $c = 0.0002975$ (low SNR), and (a) unconstrained, (b) $\alpha = 0.001$, (c) $\alpha = 0.005$, (d) $\alpha = 0.01$, (e) $\alpha = 0.02$, and (f) $\alpha = 0.05$.

scenarios is the roughness of the estimates. Additionally, when noise is placed on the squared Fourier magnitudes and the SNR is "destructively" low, noise is manifest as sinusoidal patterns.

Estimate degradation from Poisson noise was quantified via various error metrics. We observed thresholds, below which noise had only trivial effects, and over which noise "suddenly" made the estimates extremely poor.

We employed the Schulz-Snyder algorithm [93] to minimize the I-divergence, which was originally inspired by a certain EM algorithm [92]. To suppress noise artifacts, we incorporated certain types of constraints via penalties. Our implementation adapted Green's OSL algorithms, based on the theoretical fact that the Schulz-Snyder algorithm is equivalent to an EM algorithm assuming a particular Poisson data model.

In Chapter 2, we tweaked the Schulz-Snyder algorithm to make the algorithm usable for aliased autocorrelations as in x-ray crystallography. In this chapter, we also incorporated penalties in our tweaked version of the Schulz-Snyder algorithm.

Good's roughness and total variation penalties were chosen for restraining the observed noise artifacts, such as roughness. Both penalties provided nice smoothing properties. However, the textures resulting from the two penalties were quite different, especially in the background. Another important difference between the two penalties is that the total variation may preserve edges, while Good's roughness encourages smoothing of edges as well as other regions.

Interestingly, it turned out that the penalties have more sensitive effects on the estimates from aliased autocorrelations than from unaliased autocorrelations, no matter whether noise is placed on the squared Fourier magnitudes or the autocorrelations directly.

When the SNR is "destructively" low, none of the penalties could improve the estimates; the penalties always led to entirely blurred estimates that do not show any useful information.

Even though incorporating penalties helpfully improved estimate quality, there still remains a serious challenge, namely, convergence of the algorithms to local minima. We started the algorithm from the "truth" so that we could study the effect of noise on the global minima independent of local minima issues. It would also be interesting to study whether noise increases the probability of converging to an unpleasant local minimum when starting from a more generic initial estimate.

We may be able to avoid difficulties with local minima by applying some *global optimization* methods, such as simulated annealing or genetic algorithms, to the minimization of the I-divergence. This is an important avenue for future work.

CHAPTER VI

AN ITERATIVE DEAUTOCONVOLUTION ALGORITHM FOR NONNEGATIVE FUNCTIONS

6.1 Introduction

This chapter considers the inverse problem of estimating a function from its autoconvolution, i.e., the convolution of that function with itself. We refer to this as the *deautoconvolution* problem. Such problems sometimes arise in physics [27]. Due to its ill-posedness, most existing solutions to the deautoconvolution problem are based on various regularization methods such as Tikhonov's regularization method or the method of Lavrent'ev. Some analytical solutions to the deautoconvolution problem have been formulated based on such regularization methods [40, 51]. Also, by noting that an autoconvolution function is an example of a linear Volterra equation of the first kind, other approaches have been proposed [1, 2, 57, 87]. Gorenflo and Hofmann [40] studied theoretical aspects of the autoconvolution operator.

Most work on the subject has focused on analytical solutions. We instead present an iterative algorithm that tries to minimize an objective function. This work focuses specifically on the case where both the underlying function and its autoconvolution are nonnegative. We formulate an information-theoretic discrepancy measure between the observed autoconvolution data and the autoconvolution of the estimate of the desired function at the current iteration. Csiszár's *I*-divergence is used for defining such a discrepancy.

This discrepancy is minimized using the Kuhn-Tucker conditions. Because no closed-form solution is available, we find an iterative algorithm. This algorithm has a helpful set of properties. It naturally preserves support constraints, nonnegativity, and the total intensity of estimates. Such properties contribute to a proof of convergence of the difference of the two consecutive estimates of our algorithm.

6.1.1 Background on Csiszár's *I*-divergence

Csiszár's *I*-divergence is a generalization of the Kullback-Leibler distance. The Kullback-Leibler distance has appeared in various fields: statistics [38, 62], pattern recognition [53, 54, 58], and spectral analysis [97]. Until Shore and Johnson [52, 98] justified, based on their four consistency axioms, the employment of the Kullback-Leibler distance in reconstruction problems, previous justifications had counted on intuitive arguments or the information-theoretic properties of the distance measure [47]. A limitation of the Kullback-Leibler distance is that it only defines a discrepancy measure between two functions that have the same integral. To compensate for this limitation, Csiszár [23] proposed his *I*-divergence measure and extended the work of Shore and Johnson to axiomatically justify using his *I*-divergence in reconstruction problems. Unlike the Kullback-Leibler distance, Csiszár's *I*-divergence measure can accommodate cases involving two functions that have different integrals. A notable result of Csiszár's work is that, if the functions involved are nonnegative, minimizing Csiszár's *I*-divergence measure is the only choice consistent with a set of intuitive postulates such as regularity, locality, and composition-consistency.

Csiszár's results inspired much work. Snyder *et al.* [107] apply the idea of minimizing Csiszár's *I*-divergence measure to image deblurring subject to nonnegativity constraints. They proposed an iterative algorithm that gives a sequence of estimates with a nice set of properties such as guaranteed convergence to the global minimum, the nonnegativity of every estimate in the sequence, and monotonically decreasing *I*-divergence. Additionally, they argued that deterministic deblurring problems with nonnegativity constraints can be thought of as statistical estimation problems from incomplete data based on an infinite number of observed samples, using the weak law of large numbers.

An important finding in [107] may be summarized as follows. Suppose some data can be modelled as a Poisson point process, and the intensity of that process is a linear transformation of an underlying point process whose intensity we wish to estimate. Assume that infinitely many data samples are available. Then, maximizing the expected value of the loglikelihood of the Poisson data is equivalent to minimizing the I-divergence between the measured mean value of the data and the estimated mean of the data, which is an output of a linear system as stated above. This finding may be interpreted in another way. If infinitely many data samples are available, finding a maximum-likelihood solution to the problem of estimating the mean of a Poisson point process is equivalent to estimating an input from an output from a linear system with a known kernel subject to nonnegativity constraints. Such an idea is generalized and rigorously formalized by Vardi and Lee [114]. Vardi and Lee concluded that a particular problem of maximum-likelihood estimation from incomplete Poisson data is equivalent to solving a linear inverse problem subject to nonnegativity constraints. This approach has been applied to deblurring problems in computerized tomography [106].

6.1.2 Motivation

In some applications, the magnitude of the Fourier transform of an image can be measured, but not the phase. Recovering the Fourier phases from the Fourier magnitudes is equivalent to reconstructing a function from its autocorrelation, i.e. the correlation of that function with itself. Schulz and Snyder [93] used the idea of minimizing Csiszár's I-divergence measure to recover an image from its autocorrelation, and proposed the Schulz-Snyder phase retrieval algorithm. The success of their algorithm motivates applying the minimization of Csiszár's I-divergence measure to the deautoconvolution problem subject to nonnegativity constraints. As a consequence, the structure of this chapter is strongly analogous to that of [93].

6.1.3 Organization

This chapter is organized as follows. Section 6.2 gives a mathematical framework for the problem of interest by means of the I-divergence measure. The algorithm is derived and discussed briefly in Section 6.3. Sections 6.4 and 6.5 describe and prove some properties of the algorithm. Numerical examples of reconstruction of two-dimensional images are demonstrated in Section 6.6. Section 6.7 concludes our work with brief remarks.

6.2 Problem Statement

The algorithm described in this chapter can be applied to any finite-dimensional function. We develop our theory in a two-dimensional space to retain reasonable generality while remaining concise.

We first introduce definitions to be used throughout this chapter. Let $\{x(t) : t \in \mathbb{R}^2\}$ denote an input, and $\{y(s) : s \in \mathbb{R}^2\}$ denote an output produced by a nonlinear system described by

$$\int_{\mathcal{T}} x(s-t)x(t)dt = y(s), \tag{79}$$

where $\mathcal{T}$ represents the domain of t. We are interested in the case that x is real-valued and nonnegative, which ensures that y is so as well. We further assume that x has a finite support, and hence so does y. The function y is often called the *autoconvolution* of x [35, 36, 40, 51]. For numerical implementation, the functions x and y are discretized as $\{x(n) : n \in \mathcal{N}\}$ and $\{y(m) : m \in \mathcal{M}\}$, respectively, where $\mathcal{N}$ represents the two-dimensional set $\{1, 2, \ldots, N\} \times \{1, 2, \ldots, M\}$, and $\mathcal{M}$ represents a set defined as

$$\mathcal{M} \stackrel{def}{=} \{m : m = n_1 - n_2, (n_1, n_2) \in \mathcal{N}^2\}. \tag{80}$$

Then, the discretized version of the autoconvolution is as follows:

$$y(m) = \begin{cases} \displaystyle\sum_{n \in \mathcal{N}} x(n)x(m-n), & m \in \mathcal{M} \\ 0, & m \notin \mathcal{M} \end{cases}. \tag{81}$$

Our goal is to reconstruct the input $\{x(n) : n \in \mathcal{N}\}$ from the measurements $\{y(m) : m \in \mathcal{M}\}$. To formalize our problem, we define an estimate of the system output, denoted by $\hat{y}$, as

$$\hat{y}(m) = \begin{cases} \displaystyle\sum_{n \in \mathcal{N}} \hat{x}(n)\hat{x}(m-n), & m \in \mathcal{M} \\ 0, & m \notin \mathcal{M} \end{cases}, \tag{82}$$

where $\{\hat{x}(n) : n \in \mathcal{N}\}$ denotes an estimate of the input x. Using this definition of $\hat{y}$, the problem of reconstructing x can be stated as follows: provided that y is measured, find an estimate of the input $\hat{x}$ such that $\hat{y}$ is as close as possible to y in some sense. To define their closeness, we need to measure the discrepancy between $\hat{y}$ and y. Once the discrepancy measure, denoted by $I(y||\hat{y})$, is determined, our goal is to find an estimate $\hat{x}$ that minimizes the measure. Several feasible choices are available for $I(y||\hat{y})$, such as the cumulative absolute error. We are drawn to Csiszár's *I*-divergence measure [23], which is a generalization of the Kullback-Leibler distance. Csiszár's *I*-divergence is defined by

$$I(y||\hat{y}) = \sum_{m\in\mathcal{M}} \left\{ y(m) \ln \frac{y(m)}{\hat{y}(m)} + \hat{y}(m) - y(m) \right\}. \tag{83}$$

In (83), we define the following limiting quantities as

$$0 \ln \frac{0}{\alpha} \overset{def}{=} 0, \qquad 0 \ln \frac{0}{0} \overset{def}{=} 0, \qquad 0 \ln \frac{\alpha}{0} \overset{def}{=} \infty, \tag{84}$$

where α is an arbitrary positive constant. We assume that the measurement y satisfies nonnegativity (i.e., $y(m) \geq 0$ for all $m \in \mathcal{M}$).

We now formulate our problem as follows: Given a measurement y, find an $\hat{x}_0$ such that

$$\hat{x}_0 = \arg\min_{\hat{x}\geq 0} I(y||\hat{y}), \tag{85}$$

where $\hat{x} \geq 0$ means that $\hat{x}(n) \geq 0$ for all $n \in \mathcal{N}$.

6.3 Deautoconvolution Algorithm

Using fundamental calculus and the Kuhn-Tucker conditions, we obtain the necessary (but not sufficient) conditions for $\hat{x}_0$ to satisfy the condition in (85):

$$\frac{\partial I(y||\hat{y})}{\partial \hat{x}_0(n)} \begin{cases} = 0 & \hat{x}_0(n) > 0 \\ \geq 0 & \hat{x}_0(n) = 0 \end{cases}, \tag{86}$$

for all $n \in \mathcal{N}$. The first derivative of Csiszár's *I*-divergence measure can be obtained as follows:

$$\begin{aligned} &\frac{\partial I(y||\hat{y})}{\partial \hat{x}(n)} \\ &= \sum_{m\in\mathcal{M}} [\hat{x}(m-n) + \hat{x}(m-n)] + \sum_{m\in\mathcal{M}} y(m) \frac{\hat{y}(m)}{y(m)} \left[\frac{-y(m)\hat{y}'(m)}{\hat{y}(m)^2} \right] \\ &= 2 \sum_{m\in\mathcal{M}} \hat{x}(m-n) - 2 \sum_{m\in\mathcal{M}} \hat{x}(m-n) \frac{y(m)}{\hat{y}(m)}. \end{aligned} \tag{87}$$

Setting the right-hand side of the second equality in (87) equal to zero suggests the following iteration:

$$\hat{x}^{(k+1)}(n) = \hat{x}^{(k)}(n)\frac{1}{\sum_{n\in\mathcal{N}}\hat{x}^{(k)}(n)}\sum_{m\in\mathcal{M}}\hat{x}^{(k)}(m-n)\frac{y(m)}{\hat{y}^{(k)}(m)}$$
$$= \hat{x}^{(k)}(n)\frac{1}{{y_0}^{1/2}}\sum_{m\in\mathcal{M}}\hat{x}^{(k)}(m-n)\frac{y(m)}{\hat{y}^{(k)}(m)}, \tag{88}$$

where

$$y_0 \stackrel{def}{=} \sum_{m\in\mathcal{M}} y(m) = \left[\sum_{n\in\mathcal{N}}\hat{x}^{(k)}(n)\right]^2, \tag{89}$$

and $\hat{y}^{(k)}(m)$ is defined as

$$\hat{y}^{(k)}(m) = \sum_{n\in\mathcal{N}}\hat{x}^{(k)}(n)\hat{x}^{(k)}(m-n). \tag{90}$$

The relation given in (89) is proven in Property 4 below. When the algorithm is initialized, it should be satisfied that $\hat{x}^{(0)} \geq 0$ (zero divided by zero is defined as zero),

$$0 < \sum_{n\in\mathcal{N}}\hat{x}^{(0)}(n) < \infty. \tag{91}$$

The algorithm in (88) is very similar in form to the Schulz-Snyder phase retrieval algorithm [93].

6.4 Properties of Deautoconvolution Algorithm

It is desirable for estimation algorithms to incorporate known constraints, such as support or nonnegativity, on the possible solutions. Also, an iterative algorithm is hoped to produce a stable solution in the sense of [21]. Property 1–Property 3 explains how our deautoconvolution algorithm preserves nonnegativity and fixed support constraints and produces a stable solution. The proofs of the properties shown below are adapted from [93] and [21].

Property 1. (Nonnegativity)

For $k = 1, 2, \ldots,$ it holds that $\hat{x}^{(k)} \geq 0$.

Proof. Since $\hat{x}^{(0)}(n) \geq 0, \forall n \in \mathcal{N}$, $y(m) \geq 0, \forall m \in \mathcal{M}$, and $\hat{y}^{(0)}(m) \geq 0, \forall m \in \mathcal{M}$, it holds that $\hat{x}^{(1)}(n) \geq 0, \forall n \in \mathcal{N}$ by the definition in (88). By applying the same arguments for $k = 1, 2, \ldots$, it can be easily shown that $\hat{x}^{(k)} \geq 0, \forall n \in \mathcal{N}$. $\square$

Property 2. (Fixed Support)

If $\hat{x}^{(0)}(n) = 0$ for $n \in \mathcal{N}_1 \subset \mathcal{N}$, then $\hat{x}^{(k)}(n) = 0$ for $n \in \mathcal{N}_1$ and $k = 1, 2, \ldots$.

Proof. This property follows from (88). $\square$

Property 3. (Fixed Minima)

Any estimate that satisfies the Kuhn-Tucker conditions in (86) for a minimizer is a fixed point of the deautoconvolution algorithm in (88).

Proof. First, suppose that an estimate $\hat{x}^{(k)}$ satisfies the Kuhn-Tucker conditions for a minimizer given in (86) for a minimizer. Then, by the definition of the Kuhn-Tucker conditions, if $\hat{x}^{(k)}(n) > 0$ for some $n \in \mathcal{N}$, then for such n,

$$\frac{1}{{y_0}^{1/2}} \sum_m \hat{x}^{(k)}(m-n) \frac{y(m)}{\hat{y}^{(k)}(m)} = 1, \tag{92}$$

and hence $\hat{x}^{(k+1)}(n) = \hat{x}^{(k)}(n)$. If $\hat{x}^{(k)}(n) = 0$ for some $n \in \mathcal{N}$, then $\hat{x}^{(k+1)}(n) = 0$ for the n. Therefore, it holds that $\hat{x}^{(k+1)}(n) = \hat{x}^{(k)}(n), \forall n \in \mathcal{N}$. $\square$

Property 4. (Conservation of Total Intensity)

If (88) is initialized with $\hat{x}^{(0)}$ such that $\sum_{n \in \mathcal{N}} \hat{x}^{(0)}(n) = y_0^{1/2}$, then the following conditions are obtained for $k = 1, 2, \ldots$:

$$\sum_{n \in \mathcal{N}} \hat{x}^{(k)}(n) = y_0^{1/2}. \tag{93}$$

Proof. Taking summation over $n \in \mathcal{N}$ on both sides of (88), we obtain the following equalities:

$$\begin{aligned} \sum_{n \in \mathcal{N}} \hat{x}^{(k+1)}(n) &= \sum_{n \in \mathcal{N}} \hat{x}^{(k)}(n) \frac{1}{\sum_{n \in \mathcal{N}} \hat{x}^{(k)}(n)} \sum_{m \in \mathcal{M}} \hat{x}^{(k)}(m-n) \frac{y(m)}{\hat{y}^{(k)}(m)} \\ &= \frac{1}{\sum_{n \in \mathcal{N}} \hat{x}^{(k)}(n)} \sum_{m \in \mathcal{M}} \frac{y(m)}{\hat{y}^{(k)}(m)} \sum_{n \in \mathcal{N}} \hat{x}^{(k)}(n) \hat{x}^{(k)}(m-n) \\ &= \frac{1}{\sum_{n \in \mathcal{N}} \hat{x}^{(k)}(n)} \sum_{m \in \mathcal{M}} \frac{y(m)}{\hat{y}^{(k)}(m)} \hat{y}^{(k)}(m) \\ &= \frac{y_0}{\sum_{n \in \mathcal{N}} \hat{x}^{(k)}(n)}. \end{aligned} \tag{94}$$

Therefore, if the k^{th} estimate $\hat{x}^{(k)}$ satisfies

$$\sum_{n \in \mathcal{N}} \hat{x}^{(k)}(n) = y_0^{1/2}, \tag{95}$$

then it follows from (94) that

$$\sum_{n \in \mathcal{N}} \hat{x}^{(k+1)}(n) = y_0^{1/2}. \tag{96}$$

Hence, when the algorithm in (88) is initialized with an initial estimate $\hat{x}^{(0)}$ whose total intensity is $y_0^{1/2}$, then the conditions in (93) are satisfied by mathematical induction. $\square$

Property 5. (Monotonicity of *I*-divergence)

A sequence of estimates provided by (88) yields a sequence of I-divergence measure that is monotonically decreasing: $I(y||\hat{y}^{(k+1)}) \leq I(y||\hat{y}^{(k)})$*, for* $k = 1, 2, \ldots$*.*

Proof. For $k = 0, 1, 2, \ldots$, the following relations can be drawn:

$$\begin{aligned}
&I(y||\hat{y}^{(k)}) - I(y||\hat{y}^{(k+1)}) \\
&\quad= \sum_{m \in \mathcal{M}} \left[\hat{y}^{(k)}(m) - y(m)\right] + \sum_{m \in \mathcal{M}} y(m) \ln \frac{y(m)}{\hat{y}^{(k)}(m)} \\
&\qquad - \sum_{m \in \mathcal{M}} \left[\hat{y}^{(k+1)}(m) - y(m)\right] + \sum_{m \in \mathcal{M}} y(m) \ln \frac{y(m)}{\hat{y}^{(k+1)}(m)} \\
&\quad= \sum_{m \in \mathcal{M}} \left[\hat{y}^{(k)}(m) - \hat{y}^{(k+1)}(m)\right] + \sum_{m \in \mathcal{M}} y(m) \ln \frac{\hat{y}^{(k+1)}(m)}{\hat{y}^{(k)}(m)}.
\end{aligned} \tag{97}$$

Note that, by Property 4, it holds that

$$\begin{aligned}
\sum_{m \in \mathcal{M}} \hat{y}^{(k)}(m) &= \sum_{m \in \mathcal{M}} \sum_{n \in \mathcal{N}} \hat{x}^{(k)}(n) \hat{x}^{(k)}(m-n) \\
&= \sum_{n \in \mathcal{N}} \hat{x}^{(k)}(n) \sum_{n \in \mathcal{N}} \hat{x}^{(k)}(m-n) \\
&= y_0,
\end{aligned} \tag{98}$$

and similarly, $\sum_{m \in \mathcal{M}} \hat{y}^{(k+1)}(m) = y_0$. Therefore, we obtain the following equation:

$$\sum_{m \in \mathcal{M}} \left[\hat{y}^{(k)}(m) - \hat{y}^{(k+1)}(m)\right] = 0. \tag{99}$$

Using (99) and (88), the relations in (97) can be rewritten as

$$
\begin{aligned}
I(y||\hat{y}^{(k)}) - I(y||\hat{y}^{(k+1)}) &= \sum_{m\in\mathcal{M}} y(m) \ln \frac{y(m)}{\hat{y}^{(k)}(m)} \\
&= \sum_{m\in\mathcal{M}} y(m) \ln \left[\frac{\sum_{n\in\mathcal{N}} \hat{x}^{(k+1)}(n)\hat{x}^{(k+1)}(m-n)}{\hat{y}^{(k)}(m)} \right] \\
&= \sum_{m\in\mathcal{M}} y(m) \ln \sum_{n\in\mathcal{N}} \left[\frac{\hat{x}^{(k)}(n)\hat{x}^{(k)}(m-n)}{\hat{y}^{(k)}(m)} \right] \left[r^{(k)}(n) r^{(k)}(m-n) \right],
\end{aligned} \tag{100}
$$

where the last equality holds by the definition of the algorithm (88), and $r^{(k)}$ is defined by

$$
r^{(k)}(n) = \frac{1}{{y_0}^{1/2}} \sum_{m\in\mathcal{M}} \hat{x}^{(k)}(m-n) \frac{y(m)}{\hat{y}^{(k)}(m)}. \tag{101}
$$

Since the logarithm is a concave function, and it is true that

$$
\frac{\hat{x}^{(k)}(n)\hat{x}^{(k)}(m-n)}{\hat{y}^{(k)}(m)} \geq 0, \forall n; \ \sum_{n\in\mathcal{N}} \frac{\hat{x}^{(k)}(n)\hat{x}^{(k)}(m-n)}{\hat{y}^{(k)}(m)} = 1, \tag{102}
$$

we can apply Jensen's inequality [80] to (100). Then, we obtain

$$
\begin{aligned}
& I(y||\hat{y}^{(k)}) - I(y||\hat{y}^{(k+1)}) \\
&\quad \geq \sum_{m\in\mathcal{M}} y(m) \sum_{n\in\mathcal{N}} \left[\frac{\hat{x}^{(k)}(n)\hat{x}^{(k)}(m-n)}{\hat{y}^{(k)}(m)} \right] \ln \left[r^{(k)}(n) r^{(k)}(m-n) \right] \\
&\quad = \sum_{m\in\mathcal{M}} y(m) \sum_{n\in\mathcal{N}} \left[\frac{\hat{x}^{(k)}(n)\hat{x}^{(k)}(m-n)}{\hat{y}^{(k)}(m)} \right] \ln r^{(k)}(n) \\
&\qquad + \sum_{m\in\mathcal{M}} y(m) \sum_{n\in\mathcal{N}} \left[\frac{\hat{x}^{(k)}(n)\hat{x}^{(k)}(m-n)}{\hat{y}^{(k)}(m)} \right] \ln r^{(k)}(m-n) \\
&\quad = \sum_{m\in\mathcal{M}} y(m) \sum_{n\in\mathcal{N}} \left[\frac{\hat{x}^{(k)}(n)\hat{x}^{(k)}(m-n)}{\hat{y}^{(k)}(m)} \right] \ln r^{(k)}(n) \\
&\qquad + \sum_{m\in\mathcal{M}} y(m) \sum_{n'\in\mathcal{N}} \left[\frac{\hat{x}^{(k)}(m-n')\hat{x}^{(k)}(n')}{\hat{y}^{(k)}(m)} \right] \ln r^{(k)}(n') \\
&\quad = 2 \sum_{m\in\mathcal{M}} y(m) \sum_{n\in\mathcal{N}} \left[\frac{\hat{x}^{(k)}(m-n)\hat{x}^{(k)}(n)}{\hat{y}^{(k)}(m)} \right] \ln r^{(k)}(n) \\
&\quad = 2 \sum_{n\in\mathcal{N}} \hat{x}^{(k)}(n) \left[\sum_{m\in\mathcal{M}} \frac{y(m)}{\hat{y}^{(k)}(m)} \hat{x}^{(k)}(m-n) \right] \ln r^{(k)}(n) \\
&\quad = 2 y_0^{1/2} \sum_{n\in\mathcal{N}} \hat{x}^{(k)}(n) r^{(k)}(n) \ln r^{(k)}(n) \\
&\quad = 2 y_0^{1/2} \sum_{n\in\mathcal{N}} \hat{x}^{(k+1)}(n) \ln \frac{\hat{x}^{(k+1)}(n)}{\hat{x}^{(k)}(n)} = 2 y_0^{1/2} I(\hat{x}^{(k+1)}||\hat{x}^{(k)}) \geq 0,
\end{aligned} \tag{103}
$$

where $n' = m - n$, and $I(\hat{x}^{(k+1)}||\hat{x}^{(k)})$ denotes the I-divergence discrepancy between $\hat{x}^{(k+1)}$ and $\hat{x}^{(k)}$, which is nonnegative as a consequence of the strict concavity of the logarithm. Note that the rest of the terms in Csiszár's I-divergence cancel out because the integrals of $\hat{x}^{(k+1)}$ and $\hat{x}^{(k)}$ are the same. On the right-hand side in (103), the fifth and sixth equalities directly follow from the definition of the deautoconvolution algorithm. Therefore, it is proven that Csiszár's I-divergence measure is monotonically decreasing. $\square$

Corollary 3. *It holds that*

$$I(y||\hat{y}^{(k+1)}) = I(y||\hat{y}^{(k)}) \tag{104}$$

if and only if $\hat{x}^{(k)}(n) = \hat{x}^{(k+1)}(n)$ *for all* $n \in \mathcal{N}$. *This implies that* $r^{(k)}(n) = 1$ *for all* $n \in \mathcal{N}$, *satisfying* $\hat{x}^{(k)}(n) > 0$ *if and only if (104) is satisfied.*

Property 2 warns us that we should not initialize a pixel value with zero unless we have *a priori* knowledge that the pixel should be zero. This provides a convenient way of incorporating support constraints. In the absense of other information, it is desirable to initialize the algorithm with a uniform, nonzero estimate, which seems to be the most general. (We note that with the Schulz-Snyder *deautocorrelation* algorithm, it is important not to initialize the algorithm a mirror symmetric initial estimate, and hence it is customary to add a small amount of randomness to the initial uniform estimate. We do not need to worry about that in our deautoconvolution case.) As a result of Property 5, it is guaranteed that $\hat{y}^{(k)}(m)$ is always positive for those m such that $y(m)$ is positive, since if $\hat{y}^{(k)}(m)$ is zero for some m, then the algorithm will produce $I(y||\hat{y}^{(k)}) = \infty$ for those m. Hence, nonnegativity is naturally preserved.

6.5 Convergence of the Difference of Two Consecutive Estimates

This section establishes convergence of the difference of two consecutive estimates of the deautoconvolution algorithm to zero. However, this does not guarantee the convergence of the estimates to a limit point. We can show that the set of limit points is not empty, and these limit points satisfy the Kuhn-Tucker conditions, which means they are critical points. One of these critical points may be a local minimum or a saddle point. In all of our simulations, we have never observed convergence to a saddle point, but we have not proven that would always be the case in general; such explorations would involve looking at second derivatives of the I-divergence function, which we leave for future work.

Lemma 1 verifies that limit points of the sequence of estimates produced by our algorithm exist. Using this lemma, we show that a limit point of estimates must be a critical point using the preceding properties and a corollary along with the Kuhn-Tucker conditions.

When the algorithm is initialized as indicated in Property 4, the property imposes a constraint on the solution. In fact, using the property, we can reduce the feasible solution space. Let Λ be the set of functions representing this reduced solution space, *i.e.*,

$$\Lambda = \left\{ \hat{x} : \sum_{n \in \mathcal{N}} \hat{x}(n) = y_0^{1/2}, \hat{x} \geq 0 \right\}. \tag{105}$$

In addition, let Λ^* denote the set of limit points of the sequence $\{\hat{x}^{(k)}\}_{k=0}^{\infty}$ that are elements of Λ. The following lemmas and theorem will establish convergence of the difference of two consecutive estimates, $\hat{x}^{(k)}$ and $\hat{x}^{(k+1)}$. The proofs of the following lemmas and theorem are adapted from [21].

Theorem 4. (Convergence of the difference of two consecutive estimates to zero in $\mathscr{L}_1$ norm)

The sequence of the difference of two consecutive estimates $||\hat{x}^{(k+1)} - \hat{x}^{(k)}||_1$ *of the algorithm converges to zero in* $\mathscr{L}_1$ *norm.*

Proof. Because the I-divergence sequence generated by a sequence of estimates is monotonically decreasing (Property 5) and is bounded below by zero, there exists a limit $I^* \geq 0$ from the monotone convergence theorem such that [4, p. 104]:

$$\lim_{k \to \infty} I(y||\hat{y}^{(k)}) = I^*, \tag{106}$$

and moreover

$$\lim_{k \to \infty} \left\{ I(y||\hat{y}^{(k)}) - I(y||\hat{y}^{(k+1)}) \right\} = 0. \tag{107}$$

Combining (107) and (103), we obtain

$$\lim_{k\to\infty} I(\hat{x}^{(k+1)}||\hat{x}^{(k)}) = 0. \tag{108}$$

We note that the Kullback-Leibler distance is stronger than the norm $\mathscr{L}_1$ (see [63] or [22, p. 300]) in that

$$\sum_{n\in\mathcal{N}} \hat{x}^{(k+1)}(n) \ln \frac{\hat{x}^{(k+1)}(n)}{\hat{x}^{(k)}(n)} \geq \frac{1}{2\ln 2}||\hat{x}^{(k+1)} - \hat{x}^{(k)}||_1^2, \tag{109}$$

where $||\cdot||_1$ denotes the $\mathscr{L}_1$ norm (see [21, p. 299] for the definition of $\mathscr{L}_1$ norm). Since (108) is reached, the left-hand side of (109) goes to zero asymptotically. Therefore, we obtain convergence of the difference to zero in $\mathscr{L}_1$ norm:

$$\lim_{k\to\infty} \sum_{n\in\mathcal{N}} |\hat{x}^{(k+1)}(n) - \hat{x}^{(k)}(n)| = 0. \tag{110}$$

This proves the theorem. □

Lemma 1. (Properties of Set of Limit Points)

Let $\mathcal{N}$ be a finite discrete set. The set of limit points Λ^* is nonempty, compact, and connected.

Proof. Our proof employs ideas from p. 371 of [21]. We first show that Λ is closed and bounded, which means that Λ is compact. Note that, by the equality and nonnegativity constraints on Λ, we have $\Lambda \subset \left[0, y_0^{1/2}\right]^{|\mathcal{N}|}$. Hence, Λ is bounded. To show closedness of Λ, suppose it is not closed. Then, there exist $\overline{x} \in \Lambda$ and $\tilde{x} \in \Lambda$ such that $\overline{x} \in B(\tilde{x}, \epsilon)$ for an arbitrarily small $\epsilon > 0$. The open ball $B(\tilde{x}, \epsilon)$ is defined by (using notation in Moon [80])

$$B(\tilde{x}, \epsilon) = \{x \in \Lambda : ||\tilde{x} - x|| < \epsilon\}. \tag{111}$$

Now, we can select $\overline{x}$ such that

$$\overline{x}(n) = x(n) + \frac{\mathbf{1}_{\mathcal{N}}(n)\epsilon}{\sum_{n\in\mathcal{N}} \mathbf{1}_{\mathcal{N}}(n)}, \ \forall n \in \mathcal{N}, \tag{112}$$

where $\mathbf{1}_{\mathcal{N}}(n)$ denotes a uniform function whose values are 1 for all $n \in \mathcal{N}$. Consequently, since $\overline{x}$ and x are both nonnegative, we have

$$\sum_{n\in\mathcal{N}} \overline{x}(n) = \textstyle\sum_{n\in\mathcal{N}} \left[x(n) + \frac{\mathbf{1}_{\mathcal{N}}(n)\epsilon}{\displaystyle\sum_{n\in\mathcal{N}} \mathbf{1}_{\mathcal{N}}(n)} \right] = \textstyle\sum_{n\in\mathcal{N}} x(n) + \left[\frac{\epsilon \displaystyle\sum_{n\in\mathcal{N}} \mathbf{1}_{\mathcal{N}}(n)}{\displaystyle\sum_{n\in\mathcal{N}} \mathbf{1}_{\mathcal{N}}(n)} \right]$$
$$= \textstyle\sum_{n\in\mathcal{N}} x(n) + \epsilon \neq y_0^{1/2}. \tag{113}$$

This contradicts the assumption that $\overline{x} \in \Lambda$. Therefore, Λ is closed. Since Λ is closed and bounded, Λ is compact by the Heine-Borel theorem [89]. Moreover, by the Bolzano-Weierstrass theorem [4], there exist a limit point because $\hat{x}^{(k)}(n) \in \Lambda$. Thus, Λ^* is nonempty. The set of limit points is always closed. Since Λ^* is a subset of a bounded set Λ, Λ^* is also bounded. Therefore, Λ^* is compact, by the Heine-Borel theorem.

We want to show that Λ^* is connected. To do so, suppose it is disconnected. Then, there are at least two nonempty sets whose union is Λ^* that are separated by the complement of the two sets. Since Λ^* is closed, the complement is open. So, this can play a role of a disconnection. In addition, it is possible to choose a compact set C that is a subset of the complement set since the complement is nonempty and open. However, elements of the sequence $\{\hat{x}^{(k)}\}_{k=0}^{\infty}$ alternate between the two disconnections, consisting of Λ^*, infinitely many times in Λ. This implies that the compact set C is traversed by the elements of Λ infinitely many times. However, we have from Theorem 4

$$\lim_{k\to\infty} I(\hat{x}^{(k+1)}||\hat{x}^{(k)}) = 0, \tag{114}$$

and hence

$$||\hat{x}^{(k+1)} - \hat{x}^{(k)}||_1 \to 0. \tag{115}$$

Therefore, C must be visited by elements of $\{\hat{x}^{(k)}\}_{k=0}^{\infty}$ infinitely many times. However, C is compact, and thus it contains at least one limit point of $\{\hat{x}^{(k)}\}_{k=0}^{\infty}$. This contradicts the statement that $C \cap \Lambda^* \subset (\Lambda^*)^c \cap \Lambda^* = \emptyset$. Consequently, Λ^* is connected. □

Theorem 5. (Limit Points Satisfy the Kuhn-Tucker Conditions)

The limit point of $\hat{x}^{(k)}$ is a critical point i.e.,

$$\lim_{k\to\infty} \hat{x}^{(k)} = \hat{x}^*, \tag{116}$$

where $\hat{x}^*$ denotes a critical point. Since the I-divergence sequence is nonincreasing, this critical point cannot be a local maximum; it must be either a local minimum or a saddle point.

Proof. Since the solution function space is defined on a finite domain, (110) also implies pointwise convergence. By Lemma 1, existence of limit points of $\{\hat{x}^{(k)}\}_{k=0}^{\infty}$ is guaranteed. Denote a limit point of $\{\hat{x}^{(k)}\}_{k=0}^{\infty}$ as $\hat{x}^*$:

$$\lim_{k\to\infty} \hat{x}^{(k)} = \hat{x}^*. \tag{117}$$

Next, we show that $\hat{x}^*$ is a critical point. Recall that the iteration is given by $\hat{x}^{(k+1)} = \hat{x}^{(k)} r^{(k)}$. If we take limits of the both sides of this equation, then we obtain

$$\hat{x}^* = \hat{x}^* r^*, \tag{118}$$

where r^* denotes a limit point of $r^{(k)}$. Existence of $\hat{x}^*$ implies existence of r^*, since if r^* diverges, then $\hat{x}^*$ diverges as well. Note that, if $\hat{x}^*$ is nonzero, r^* must be one to guarantee consistency of (118). If $\hat{x}^*$ is zero, then we should have infinitely many $r^{(k)} \leq 1$ as k goes to infinity. Note that

$$\hat{x}^{(k)}(n) = \hat{x}^{(0)}(n) \prod_{i=0}^{k} r^{(i)}(n), \ \forall n \in \mathcal{N}. \tag{119}$$

The right-hand side of (119) would diverge when k goes to infinity, unless we have infinitely many $r^{(k)} \leq 1$. Therefore, the limit of $r^{(k)}$ satisfies that $r^* \leq 1$. Consequently, we obtain

$$r^* = \begin{cases} = 1 \ \hat{x}^* > 0 \\ \leq 1 \ \hat{x}^* = 0 \end{cases} \tag{120}$$

Therefore, the Kuhn-Tucker conditions given in (86) are satisfied, and hence $\hat{x}^*$ is a critical point. □

We emphasize that the convergence of the difference between estimates does not guarantee the convergence of the estimates of the algorithm. The proof of convergence of the estimates remains as an important future research. Another question would involve the uniqueness (or lack thereof) of the limit point. Such a proof might follow along the lines of [21]. However, [21] involves a linear system with a fixed kernel; in our autoconvolution case, the equivalent kernel changes with each iteration. As a result, Cover's "alternating minimization" arguments would need to be extended. This will cause some complications in asserting the uniqueness of the limit point of the sequence produced by the deautoconvolution algorithm. So, a proof of uniqueness of the limit (if one is available) remains for future work.

6.6 Numerical Examples

The preceding sections set up the mathematical foundation of our deautoconvolution algorithm. This section shows examples of some images reconstructed from their autoconvolution data. As mentioned before, we adhere to examples of two-dimensional images. The algorithm is implemented by the following sequence of steps:

1. Begin with an input estimate $\hat{x}^{(0)}$ that is a valid image estimate (nonnegative and normalized according to Eq. (93)).

2. Convolve $\hat{x}^{(k)}$ with itself to obtain $\hat{y}^{(k)}$.

3. Divide the measurement y by the estimated output $\hat{y}^{(k)}$. Call this function $u^{(k)}$.

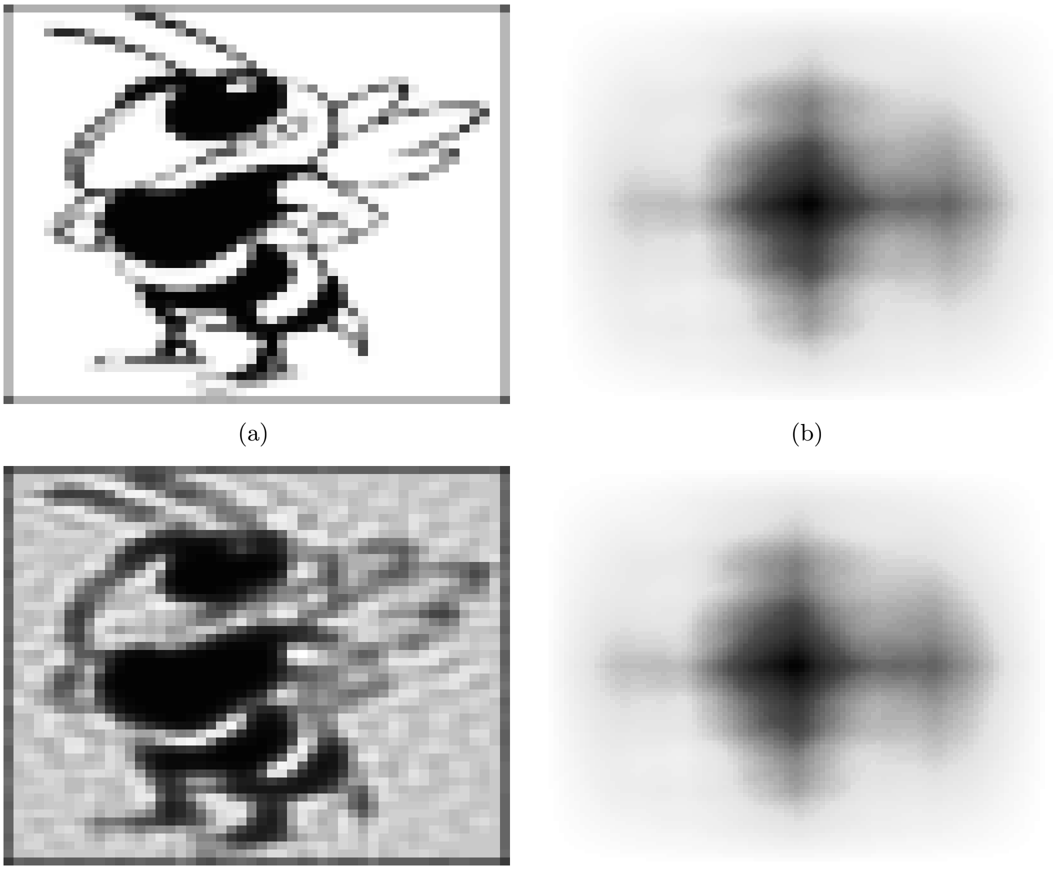
4. Compute $r^{(k)}(n) = \frac{1}{y_0^{1/2}} \sum_{m \in \mathcal{M}} \hat{x}^{(k)}(m-n) u^{(k)}(m)$.

5. Update the estimate of $\hat{x}^{(k)}$ by

$$\hat{x}^{(k+1)}(n) = \hat{x}^{(k)}(n) r^{(k)}(n), \ \forall n \in \mathcal{N}. \tag{121}$$

6. Repeat steps 2 through 4 until a convergence criterion is met.

Figure 82 demonstrates the reconstruction of a two-dimensional image. The size of the two-dimensional image is 50×50 pixels. Figures 82(a) and 82(c) show a two-dimensional original image and an estimate of the image provided by the algorithm at the 20000-*th* iteration. The original image possesses various interesting details such as the feelers and wings. Figures 82(b) and 82(d) show the autoconvolutions of 82(a) and 82(c), respectively. In Figure 82, the colormaps of the yellow jacket images (we have chosen "Buzz," the mascot

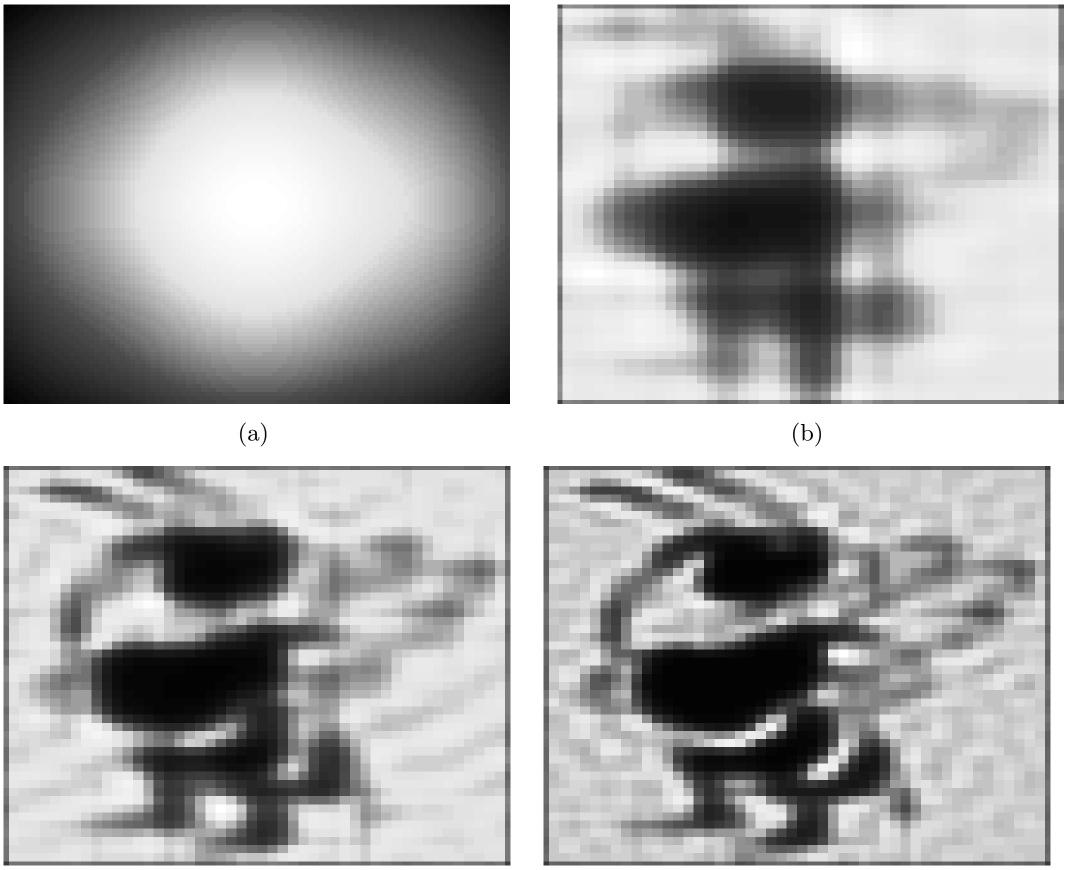


(a) (b) (c) (d)

Figure 82: (a) Original image used in numerical experiments. (b) Autocorrelation of the original image. (c) Image estimate at the 20000-*th* iteration. (d) Autocorrelation of the image estimate

of the authors' institution) and the autoconvolutions are different, to best display features. For the images, black represents low values, and white represents high values; for the autocorrelations, black represents high values, and white represents low values. The estimate is remarkable in that the autoconvolution image does not show any resemblance to the original image. However, the estimate looks quite similar to the original image and shows most of details that the original image shows. The autoconvolution images look quite similar to each other as well.

Figure 83 shows some interesting, intermediate reconstructions. Our algorithm is initialized with a constant estimate, where all the values are the same and appropriately scaled. Note how different the estimates at the first iteration and at the 15000-*th* iteration are.



(a) (b) (c) (d)

Figure 83: Selected reconstructions of Figure 82(a) at the 1-*st* (a), 500-*th* (b), 5000-th (c), and 15000-th (d) iteration.

At the 5000-*th* iteration, the yellow jacket is already somewhat identifiable. Although the solution at the 5000-*th* iteration is usable, we run the algorithm to the 20000-*th* iteration until changes in the estimate are hardly observable.

Figures 84(a) and 84(c) show a second test image, created by extracting edges from the yellow jacket, and an estimate of this image after 1000 iterations of the deautoconvolution algorithm. Notice that the estimate looks almost the same as the edge-extracted image. The autoconvolutions of them are almost the same as well. Here, in Figure 84, the colormaps for the original images and the autoconvolutions are the same (unlike in the previous example). Black represents high values, and white represents low values. Figures 84(b) and 84(d) show the autoconvolution images of Figures 84(a) and 84(c), respectively. It is interesting that the estimates of the edge-extracted image converge much faster than the original image in

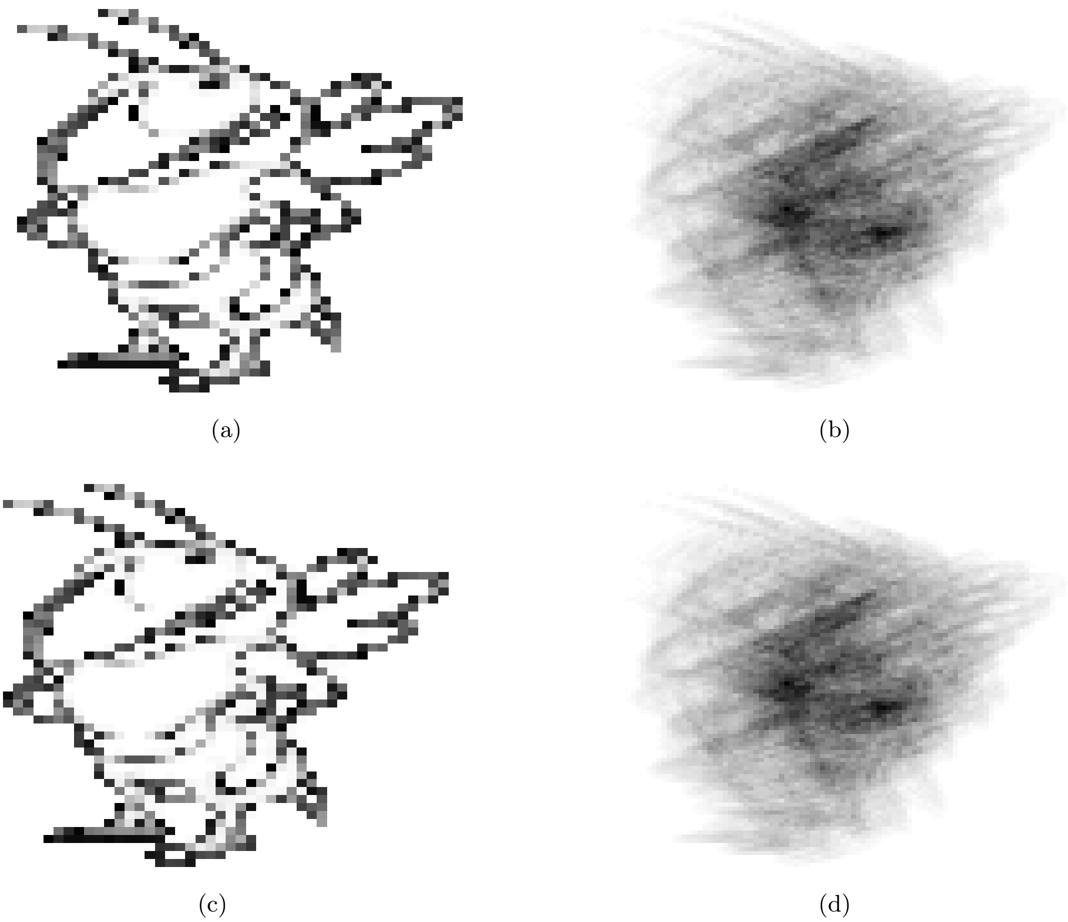


Figure 84: (a) Original image used in numerical experiments. (b) Autocorrelation of the original image. (c) Image estimate at the 1000-*th* iteration. (d) Autocorrelation of the image estimate

Figure 82(c). Noting that we started with the same initial estimate, we might conjecture that the dimension of the space of nonzero-valued parameters in the image affects the speed of convergence. Figure 85 shows some selected iterations. As in Figure 83, the estimates in the earlier stage look like blurred versions of the estimates in the later stage. As before, the algorithm is run to the 1000-*th* iteration to obtain a "visibly best" image.

6.7 Conclusions

We have proposed a *deautoconvolution* algorithm that estimates a nonnegative function from its autoconvolution. Since our deautoconvolution algorithm has basically the same

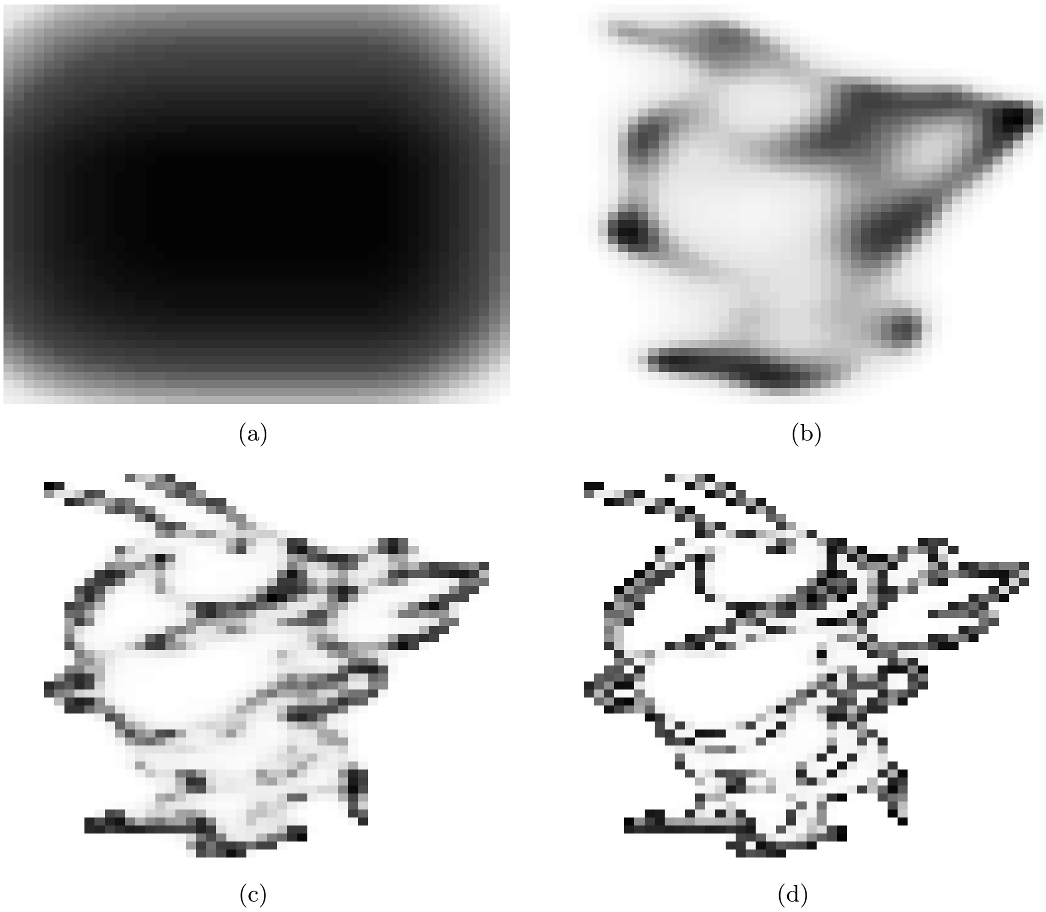


Figure 85: Selected reconstructions of Figure 84(a) at the 1-*st* (a), 70-*th* (b), 300-*th*, and 700-*th* iteration.

mathematical foundation as the Schulz-Snyder phase retrieval algorithm and Cover's algorithm for maximizing log-investment return of a portfolio, most of our mathematical proofs are based on their work [21, 93].

The algorithm naturally incorporates constraints on the solution such as nonnegativity and known image support. The algorithm also possesses other nice properties such as guaranteed monotonically decreasing I-divergence and conservation of total intensity, which implies a reduced space of solutions. Furthermore, convergence of the difference of two consecutive estimates of the algorithm to zero has been shown. Also, we have analytically shown that a limit point of the estimates of the algorithm is a critical point (either a local minimum or a saddle point). Although we might conjecture that the algorithm will not suffer from convergence to saddle points based on our experiments, a proof of such a conjecture (if it exists) remains for future discovery. Additional questions remain about the possibility of the algorithm becoming trapped in local, but not global, minima, as in the case of the Schulz-Snyder phase retrieval algorithm (see Chapter 4). We do not know if such local minima of the I-divergence surface exist. We have not encountered any in our experiments, but that does not prove that they will never be there. This also requires further analysis.

Results from the numerical experiments are promising. The solutions provided by the deautoconvolution images are inspiringly close to the original images. Even though we do not show experiments where the measured data are corrupted by noise, our experience is that the algorithm is still robust to such cases. Studies with different levels of noise remain an avenue for future work.

CHAPTER VII

PENALIZED MINIMUM I-DIVERGENCE METHODS FOR THE INVERSE BLACKBODY RADIATION PROBLEM

7.1 Introduction

A blackbody is a theoretical object that completely absorbs all wavelengths of thermal radiation incident on it. Consequently, it does not reflect light and appears black, unless the object itself radiates because of its high temperature. When a blackbody is heated to a particular temperature, it emits thermal radiation with the maximum amount of energy possible for that temperature. This phenomenon is known as blackbody radiation. Even though a blackbody is an ideal object, some materials, such as carbon in its graphite form, act like a blackbody. For instance, carbon absorbs about 97 percent of incident radiation and also acts as a perfect emitter of radiation [60].

The inverse blackbody radiation problem is to find the area temperature distribution $a(T)$ of a blackbody from the total radiated power spectrum $W(\nu)$ radiated from the blackbody. $W(\nu)$ is induced by the integral equation specified by Planck's law:

$$W(\nu) = \frac{2h\nu^3}{c^2} \int_0^\infty \frac{1}{\exp\left(\frac{h\nu}{kT}\right) - 1} a(T) dT, \tag{122}$$

where ν is frequency, T absolute temperature, h Planck's constant, k Boltzmann's constant, and c the speed of light. The integral kernel, called *spectral brightness*, is given by

$$P(T, \nu) = \frac{2h\nu^3}{c^2} \frac{1}{\exp\left(\frac{h\nu}{kT}\right) - 1}. \tag{123}$$

Since Bojarski [7] first brought up this problem, various solutions have been proposed [8,18,28,44,48,61,76,109,112,117]. The earliest methods [8,44,48,61] are practically unstable because they exploit the inverse Laplace transform, which is known to be ill-posed. Hence, as noted by Dou and Hodgson [28], such solutions often fail to produce reasonable solutions with real data, which always involve noise. Some recently proposed methods [28,76,109,112] alleviate this instability and have shown success in producing "reasonably good" estimates.

Even when the inverse Laplace transform is not explicitly used, the inherent ill-posedness remains due to the characteristics of the kernel $P(T, \nu)$. This ill-posedness has been thoroughly analyzed by Sun and Jaggard [109]. They noted that the kernel of the integral equation in Eq. (122), $P(T, \nu)$, shows the following limiting behavior:

$$\begin{aligned} &\lim_{T \to 0} P(T, \nu) \to 0, \\ &\lim_{T \to \infty} P(T, \nu) \to \infty, \end{aligned}$$

for a fixed ν. This implies that the high-temperature contributions of $a(T)$ in $W(\nu)$ dominate; hence, much of the low-temperature distribution information is lost.

Addressing this ill-posedness, some experiments showed the methods proposed by Sun and Jaggard [109] and Dou and Hodgson [28] produced stable solutions. Dou and Hodgson remarked on the practical implementation burden of the method by Sun and Jaggard and proposed a maximum-entropy method. However, their maximum-entropy method is also complicated.

When the total radiated power spectrum measurements are corrupted by noise, simple solutions may greatly suffer. The regularization method proposed by Sun and Jaggard combats this problem and attains somewhat reasonable solutions. Tan *et al.* [112] used a hybrid input-output projection algorithm to find noise-robust solutions.

Due to the problem's inherent ill-posedness, there are limitations on algorithm performance, especially when noisy measurements are involved. In extreme cases, the problem cannot be "satisfactorily" solved no matter how smart the algorithm is designed. There does not yet appear to be any work that discusses such fundamental limits in detail.

Our work is highly inspired by the regularization method by Sun and Jaggard [109] and the maximum entropy method by Dou and Hodgson [28]. However, our methods are fundamentally different from theirs in that our methods define data consistency based on an information-theoretic discrepancy measure called Csiszár's *I*-divergence, instead of the squared-error measure. We incorporate regularization via various penalties, including an entropy-based term. We discuss the fundamental limitations of our estimation methods based on an analysis of the ill-posedness of the inverse blackbody radiation problem.

Csiszár's I-divergence [23] (also called cross entropy in the related literature [10, 11, 74]) is a discrepancy measure defined on two nonnegative functions. It can be thought of as a generalization of the Kullback-Leibler distance [64]. A notable result of Csiszár's work [23] is that, if the functions involved are nonnegative, minimizing Csiszár's I-divergence measure is the only choice consistent with a set of intuitive postulates such as regularity, locality, and composition-consistency, which are desirable for estimation problems. Our work was prompted by the observation that all the functions involved in the inverse blackbody radiation problem are nonnegative.

Snyder *et al.* [105, 107] found that maximizing the expected value of the loglikelihood of Poisson data is equivalent to minimizing I-divergence between the measured mean value of the data and a hypothetical data mean derived from a function of interest through a linear mapping. Noting this relationship, Snyder, Schulz, and O'Sullivan [107] proposed an iterative method based on minimizing the I-divergence to address deblurring problems subject to nonnegativity constraints. The inverse blackbody radiation problem can be viewed as a "shift-variant" deblurring problem where the area temperature distribution is blurred by $P(T, \nu)$. Vardi and Lee [114] offer an alternative interpretation to that of Snyder *et al.* In their interpretation, they imagine the measured data are independent identically distributed (*i.i.d.*) samples $Y_j, j = 1, 2, \ldots, J$, of a random variable Y, with probability mass function $p(Y = Y_j) = y_j$, derived from hypothetical *i.i.d.* data samples $X_i, i = 1, 2, \ldots, I$, of a random variable X with probability mass function $p(X = X_i) = x_i$, and the transition probability mass function is $p(Y = Y_j | X = X_i) = h_{ij}$. Hence, the measured "incomplete data" are related to hypothetical "complete data"[1] via

$$y_j = \sum_i h_{ij} x_i. \tag{124}$$

An important result of their work is that the algorithms that seek to minimize the I-divergence can be interpreted as expectation-maximization (EM) algorithms corresponding

[1]The terminology, complete data and incomplete data, often confuses people outside the field of statistics. The choice of the terminology originally comes from statistics problems pertaining to missing data.

to the data model given in Eq. (124).

When measurements do not contain noise, and the ill-posedness is not too strong, our unconstrained methods compare favorably with the other methods cited earlier. However, once the measurements are corrupted by noise and/or the ill-posedness becomes severe, unconstrained methods do not perform well. Hence, we regularize our estimates using penalties. Similar formulations and related solution methods can be found in [84].

Since we have observed certain types of undesirable artifacts in unconstrained (*i.e.*, unregularized) estimates, we choose penalties that restrain such artifacts. We explore Shannon's entropy, the L_1 norm, and Good's roughness. Penalizing estimates by maximizing entropy subject to data fidelity constraints has been popular [3, 26, 65, 79] for inverse problems described by the Fredholm equation of the first kind. An important property of maximum-entropy regularization is the shrinkage of estimates on a component basis [26]. Since the L_1-norm penalty also shows shrinkage behavior [26], we consider the L_1-norm penalty as well.

Good's roughness penalty [39] has found success in regularizing estimates in emission tomography [67]. Our unconstrained algorithm has the same form as the EM algorithm for the Poisson data model used in emission tomography. Hence, we investigate the effects of Good's roughness penalty in our algorithms.

Another reason for considering L_1-norm and Good's roughness penalties is that they may be thought of as analogous to the energy-based regularization terms used in [109].

Once a penalty is involved, the pertinent optimization performed at each iteration is, in general, no longer simple. Green proposed the so-called one-step-late (OSL) algorithms [41] to resolve this complication in EM algorithms. Using the relationship between the minimum I-divergence algorithms and the corresponding EM algorithms mentioned earlier, we adapt Green's OSL algorithms to solve the implementation problems we encountered.

In addition to suppressing undesirable artifacts, regularizing estimates by a penalty provides faster convergence. This behavior of penalized estimates has been investigated in emission tomography [31]. Green also provides a brief theoretical discussion on this behavior of penalized estimates in view of his OSL algorithms [41].

This chapter is organized as follows. Section 7.2 formulates an unconstrained minimum I-divergence algorithm. Section 7.3 discusses penalized minimum I-divergence algorithms along with specific penalties and relevant optimization methods. The effectiveness of these methods is illustrated via various numerical experiments in Section 7.4. We finally conclude our discussion and suggest possibilities for future work in Section 7.5.

7.2 An Unconstrained Minimum I-divergence Method

We formulate the inverse blackbody problem in an optimization framework. Our goal is to find an area-temperature distribution $\hat{a}$ such that the total radiated power spectrum $\hat{W}$ that would result from $\hat{a}$, via the integral equation described by Planck's law, is "closest" to the measured total radiated power spectrum W in the sense of some discrepancy measure. Our choice for such a discrepancy measure is Csiszár's I-divergence. Restating the problem more formally, our goal is to find $\hat{a}_0$ such that

$$\begin{aligned} \hat{a}_0 &= \arg\min_{\hat{a}\geq 0} I(W||\hat{W}), \\ &= \arg\min_{\hat{a}\geq 0} \int \left\{ W(\nu)\log\frac{W(\nu)}{\hat{W}(\nu)} - W(\nu) + \hat{W}(\nu) \right\} d\nu, \end{aligned} \tag{125}$$

where $\hat{a} \geq 0$ means that all components of $\hat{a}$ are nonnegative, and the total radiated power spectrum $\hat{W}$, associated with $\hat{a}$, is given by

$$\hat{W}(\nu) = \frac{2h\nu^3}{c^2} \int_0^\infty \frac{1}{\exp\left(\frac{h\nu}{kT}\right) - 1} \hat{a}(T) dT. \tag{126}$$

Using the Kuhn-Tucker conditions [73], we obtain the following necessary (but not sufficient) conditions for $\hat{a}_0$ to achieve Eq. (125):

$$\frac{\partial I(W||\hat{W})}{\partial \hat{a}_0(T)} \begin{cases} = 0, \ \hat{a}_0(T) > 0 \\ \geq 0, \ \hat{a}_0(T) = 0 \end{cases}. \tag{127}$$

Let $P(T,\nu)$ be the kernel of the integral equation that produces $\hat{W}$, namely Eq. (123).

Now, we consider discretizations of ν and T to allow for implementation on a digital computer such that the integral kernel, the power spectrum W, and the I-divergence become

$$P(T_i, \nu_j) = \frac{2h\nu_j^3}{c^2} \frac{1}{\exp\left(\frac{h\nu_j}{kT_i}\right) - 1}, \tag{128}$$

$$W(\nu_j) = \sum_i P(T_i, \nu_j) a(T_i),$$

$$I(W||\hat{W}) = \sum_j \left\{ W(\nu_j) \log \frac{W(\nu_j)}{\hat{W}(\nu_j)} - W(\nu_j) + \hat{W}(\nu_j) \right\}. \tag{129}$$

For this discretization, the first derivative of the I-divergence can be obtained as follows:

$$\begin{aligned} \frac{\partial I(W||\hat{W})}{\partial \hat{a}(T_i)} &= \sum_j \left\{ W(\nu_j) \frac{\hat{W}(\nu_j)}{W(\nu_j)} \frac{-W(\nu_j)}{[\hat{W}(\nu_j)]^2} \frac{\partial \hat{W}(\nu_j)}{\partial \hat{a}(T_i)} + \frac{\partial \hat{W}(\nu_j)}{\partial \hat{a}(T_i)} \right\} \\ &= \sum_j \left\{ \frac{-W(\nu_j)}{\hat{W}(\nu_j)} \frac{\partial \hat{W}(\nu_j)}{\partial \hat{a}(T_i)} + \frac{\partial \hat{W}(\nu_j)}{\partial \hat{a}(T_i)} \right\}. \end{aligned} \tag{130}$$

The derivative of $\hat{W}$ is

$$\frac{\partial \hat{W}(\nu_j)}{\partial \hat{a}(T_i)} = \frac{\partial}{\partial \hat{a}(T_i)} \sum_n P(T_n, \nu_j) \hat{a}(T_n) = P(T_i, \nu_j). \tag{131}$$

Therefore, the derivative in Eq. (130) becomes

$$\frac{\partial I(W||\hat{W})}{\partial \hat{a}(T_i)} = \sum_j \left\{ \frac{-W(\nu_j)}{\hat{W}(\nu_j)} P(T_i, \nu_j) + P(T_i, \nu_j) \right\}. \tag{132}$$

We are drawn to a multiplicative, iterative algorithm that is suggested by setting Eq. (132) equal to zero:

$$\hat{a}^{(k+1)}(T_i) = \hat{a}^{(k)}(T_i) \frac{1}{\phi(T_i)} \sum_j P(T_i, \nu_j) \frac{W(\nu_j)}{\hat{W}^{(k)}(\nu_j)}, \tag{133}$$

where $\phi(T_i)$ and $\hat{W}^{(k)}(\nu_j)$ are given by

$$\phi(T_i) = \sum_j P(T_i, \nu_j), \tag{134}$$

$$\hat{W}^{(k)}(\nu_j) = \sum_i P(T_i, \nu_j) \hat{a}^{(k)}(T_i). \tag{135}$$

Of course, one could imagine other potential $\hat{a}^{(k)}$ and $\hat{a}^{(k+1)}$ arrangements. The choice in Eq. (133) is given more justification in Section 7.3.2.

The algorithm in Eq. (133) can be viewed as the deblurring algorithm proposed by Snyder *et al.* [107] when the blurring kernel is P, which is characterized by Planck's law in this particular problem. As discussed in [107], this algorithm possesses various desirable properties such as guaranteed convergence to the global minimum (under mild conditions) and monotonically decreasing I-divergence.

7.3 Penalized Minimum I-divergence Methods

The unconstrained minimum I-divergence algorithm given in Eq. (133) provides reasonable estimates (compared with other existing methods) when the total radiated power spectrum measurements are not corrupted by noise. However, real measurements are always corrupted by noise in practice. With noisy measurements, the unconstrained algorithm often severely suffers from the ill-posedness of the kernel [109] of the integral equation. Many ill-posed estimation problems have been successfully addressed by regularization methods. For the inverse blackbody radiation problem, we consider three different penalties for suppressing undesirable artifacts in unconstrained reconstructions from noisy measurements: L_1-norm [25], entropy [26], and Good's roughness [39], or equivalently, O'Sullivan's roughness penalty [83].

The idea of regularization leads to new optimization formulation, where we aim to find $\hat{a}$ such that

$$\hat{a}_0 = \arg\min_{\hat{a}\geq 0} I(W||\hat{W}) + \lambda\Phi(\hat{a}), \tag{136}$$

where λ is a regularization parameter that determines how influential the penalty term $\Phi(\hat{a})$ will be on the estimates. We may alternatively consider the following formulation:

$$\hat{a}_0 = \arg\min_{\hat{a}\geq 0} I(W||\hat{W}) + \Phi(\hat{a};\alpha), \tag{137}$$

where α is a regularization parameter vector, and the penalty $\Phi(\hat{a};\alpha)$ is defined by

$$\Phi(\hat{a};\alpha) = \sum_i \alpha_i f_i(\hat{a}), \tag{138}$$

where f_i is a function that depends on the penalty type. For our study, we consider the second formulation because it helps address the characteristics of the kernel of the integral equation by letting us control the influence of the penalty individually for each component as well as avoid numerical difficulties.

7.3.1 Discussion on Penalties

When noisy measurements enter the unconstrained minimum I-divergence algorithm, the algorithm is highly inclined to produce estimates with excessively large peaks that seem to dominate all other components in the estimate. This motivates us to choose certain penalties to suppress such undesirable artifacts.

As well studied in [26], L_1-norm and maximum-entropy penalties have inherent shrinkage properties. In particular, maximum-entropy encourages the estimate to shrink towards a nominal value of $1/e$ componentwise. Donoho *et al.* [26] provide an excellent discussion on the practical behavior of entropy regularization. Lanterman [68] discusses entropy regularization in the context of radio astronomy. Another nice discussion about L_1-norm regularization can be found in [25].

For entropy-like regularization, the penalty term in Eq. (137) is defined by

$$\Phi_E(\hat{a};\alpha) = \sum_i \alpha_i \hat{a}(T_i) \log \hat{a}(T_i). \tag{139}$$

Note that since we want to encourage maximization of entropy, which is $-\Phi(\hat{a};\alpha)$, we equivalently encourage the minimization of the negated entropy.

For L_1-norm regularization, the penalty is defined as

$$\Phi_{L_1}(\hat{a};\alpha) = \sum_i \alpha_i \hat{a}(T_i). \tag{140}$$

Another choice is Good's roughness penalty [39]. This penalty characterizes the roughness of an estimate by minimizing a quantity related to the energy in the first derivative of the estimate, which in the continuous domain is

$$\Phi_G(\hat{a};\alpha) = \int \alpha(T)\hat{a}(T) \left[\frac{\partial \log \hat{a}(T)}{\partial T}\right]^2 dT \tag{141}$$

$$= \int \alpha(T) \frac{1}{\hat{a}(T)} \left[\frac{\partial \hat{a}(T)}{\partial T}\right]^2 dT, \tag{142}$$

where $\alpha(T)$ represents our regularization parameter function, which in previous applications has been set to a constant. We describe it as a function to stay consistent with our formulation given earlier. Discretization of Eq. (141) yields

$$\Phi_G(\hat{a};\alpha) = \sum_i \alpha_i \hat{a}(T_i)\{2\log \hat{a}(T_i) - \log \hat{a}(T_{i-1}) - \log \hat{a}(T_{i+1})\}. \tag{143}$$

O'Sullivan [83] noted that this discretized version of the penalty is equivalent to his roughness penalty on finite domains expressed in terms of Csiszár's I-divergences between neighboring components:

$$\Phi_G(\hat{a};\alpha) = \sum_i \alpha_i \left\{ I(\hat{a}||S_l\hat{a}) + I(\hat{a}||S_r\hat{a}) \right\}, \tag{144}$$

where $S_l\hat{a}$ and $S_r\hat{a}$ are defined by

$$\begin{aligned} \left[S_l\hat{a}\right](T_i) &= \hat{a}(T_{((i+1) \bmod n)}), \\ \left[S_r\hat{a}\right](T_i) &= \hat{a}(T_{((i-1) \bmod n)}), \end{aligned} \tag{145}$$

where n represents the length of $\hat{a}$. Note that the components are circularly shifted.

We expected that O'Sullivan's roughness penalty, as indicated by its definition, would suppress the undesirable, large peaks by reducing the difference between the peaks and their neighboring components, which would be relatively much smaller.

7.3.2 A Bridge between EM Algorithms and Minimum I-divergence Algorithms

Recall that our goal is to find the $\hat{a}_0$ that achieves

$$\hat{a}_0 = \arg\min_{\hat{a}\geq 0} I(W||\hat{W}) + \Phi(\hat{a};\alpha). \tag{146}$$

Note the following relations:

$$\begin{aligned} &\arg\min_{\hat{a}\geq 0} I(W||\hat{W}) + \Phi(\hat{a};\alpha) \\ &= \arg\min_{\hat{a}\geq 0} \sum_j \left\{ W(\nu_j)\log\frac{W(\nu_j)}{\hat{W}(\nu_j)} - W(\nu_j) + \hat{W}(\nu_j) \right\} + \Phi(\hat{a};\alpha) \\ &= \arg\min_{\hat{a}\geq 0} \sum_j \left\{ W(\nu_j)\log W(\nu_j) - W(\nu_j) \right\} - \sum_j \left\{ W(\nu_j)\log\hat{W}(\nu_j) - \hat{W}(\nu_j) \right\} + \Phi(\hat{a};\alpha) \\ &= \arg\max_{\hat{a}\geq 0} \sum_j \left\{ W(\nu_j)\log\hat{W}(\nu_j) - \hat{W}(\nu_j) \right\} - \Phi(\hat{a};\alpha), \end{aligned} \tag{147}$$

where the last equality holds since the term $\sum_j \left[W(\nu_j)\log W(\nu_j) - W(\nu_j)\right]$ does not depend on $\hat{a}$. It is remarkable that the last line in Eq. (147) can be interpreted as maximum penalized-likelihood estimation assuming a Poisson data model [105,107], or more generally, the linearly related incomplete-complete data model [114]. This interpretation informs us that a sequence of $\hat{a}$ that can achieve maximum penalized-likelihood can also achieve minimum penalized-I-divergence.

Expectation-Maximization (EM) algorithms are strategic methods for producing a sequence of estimates $\hat{a}^{(k)}$ that attempt to maximize the penalized likelihood by solving for $\hat{a}^{(new)}$

$$D^{10}Q(\hat{a}^{(new)}|\hat{a}^{(old)}) - D\Phi(\hat{a}^{(new)};\alpha) = 0, \tag{148}$$

where

$$Q(\hat{a}^{(new)}|\hat{a}^{(old)}) = E\left[\log p(X|\hat{a}^{(new)})|W, \hat{a}^{(old)}\right]. \tag{149}$$

In the formulas above, D denotes the derivative operator with respect to the parameters involved (*e.g.*, $D^{10}Q(\hat{a}^{(new)}|\hat{a}^{(old)})$ denotes the first-order partial derivative of Q with respect to $\hat{a}^{(new)}$), Q is the expectation of the loglikelihood $\log p(X|\hat{a}^{(new)})$ of hypothetical "complete data" X given the current estimate of the parameter $\hat{a}^{(old)}$, and the measured "incomplete data" W. Here, the hypothetical complete data are assumed to follow a Poisson distribution:

$$X(T_i, \nu_j) \sim Poisson(a(T_i)P(T_i, \nu_j)), \tag{150}$$

and the incomplete data and the complete data are related by

$$W(\nu_j) = \sum_i X(T_i, \nu_j). \tag{151}$$

For the complete description of this setting and notation, readers may refer to Green's original work [41] and the work by Dempster *et al.* [24].

Therefore, exactly the same sequence $\hat{a}^{(k)}$ generated by EM algorithms may be used to minimize the penalized I-divergence, provided that the EM algorithms are designed by assuming the incomplete-complete data model under a linear relation. Section 7.3.4 uses this theoretical connection to adapt Green's OSL methods to the penalized I-divergence optimization problem given in Eq. (137).

7.3.3 Optimization Challenge

Recall that the penalized EM algorithms, assuming the linearly-related incomplete-complete data model [114], can maximize the penalized-likelihood described earlier and equivalently minimize the penalized-I-divergence. The appropriate penalized EM algorithm for the data model finds the $\hat{a}$ which maximizes the penalized complete-data loglikelihood [94]

$$\begin{aligned} & Q(\hat{a}|\hat{a}^{(old)}) - \Phi(\hat{a};\alpha) \\ & = \sum_i \sum_j \left[-\hat{a}(T_i)P(T_i,\nu_j) + \hat{a}^{(uc)}(T_i) \log\left\{\hat{a}(T_i)P(T_i,\nu_j)\right\} \right] - \Phi(\hat{a};\alpha), \end{aligned} \tag{152}$$

at each iteration. In Eq. (152), $\hat{a}^{(old)}$ represents the estimate at the previous iteration, and $\hat{a}^{(uc)}$ represents the estimates produced by the unconstrained (or unregularized) EM iteration, which is the same as the unconstrained minimum I-divergence algorithm given in Eq. (133). Note that if no penalty is present, setting the derivative of Eq. (152) with respect to $\hat{a}$ to zero yields a closed-form solution for $\hat{a}$, yielding Eq. (132). When a penalty is introduced, the optimization problems involving a penalty are no longer trivial. In particular, components of $\hat{a}$ become coupled with their neighboring components in the optimization involving Good's roughness:

$$\begin{aligned} & Q(\hat{a}|\hat{a}^{(old)}) - \Phi_G(\hat{a};\alpha) \\ & = \sum_i \sum_j \left[-\hat{a}(T_i)P(T_i,\nu_j) + \hat{a}^{(uc)}(T_i) \log\left\{\hat{a}(T_i)P(T_i,\nu_j)\right\} \right] \\ & \quad - \sum_i \alpha_i \hat{a}(T_i)\{2\log\hat{a}(T_i) - \log\hat{a}(T_{i-1}) - \log\hat{a}(T_{i+1})\}. \end{aligned} \tag{153}$$

Several methods, such as gradient-based methods [56, 75, 104], can be used to solve this non-trivial optimization problems. One alternative that is easy to apply and implement is Green's one-step-late (OSL) algorithm. We discussed an important relationship between minimum penalized-I-divergence algorithms and penalized EM algorithms earlier. This serves as a foundation of how the OSL algorithms can be adapted to our framework.

7.3.4 Green's One-step-late (OSL) Algorithms

Green's OSL algorithms were originally tweaks of EM algorithms intended for maximum penalized-likelihood estimation. Green [41] noted that the pertinent objective function in an EM algorithm can be linearized at the current estimate as in gradient methods, and the derivatives of the penalty term at two consecutive iterations have a small difference if the algorithm has slow convergence, as is the case with EM algorithms [24] and other multiplicative algorithms [66]. In an EM formulation, these observations motivate finding a parameter θ that solves

$$D^{10}Q(\hat{a}^{(new)}|\hat{a}^{(old)}) - D\Phi(\hat{a}^{(old)};\alpha) = 0. \tag{154}$$

Note that Eqs. (154) and (148) differ only in that Eq. (154) uses $\hat{a}^{(old)}$ in the penalty term, while (148) uses $\hat{a}^{(new)}$. An appealing property of this algorithm is that Eqs. (154) and (148) have the same fixed points.

Green's OSL algorithms have empirically shown monotonically increasing penalized-likelihood (of incomplete data) and show faster convergence behavior compared to the corresponding unconstrained EM algorithms. However, we emphasize that the faster convergence behavior is due to the penalty, rather than properties of the OSL-algorithm. Another advantage of using the OSL idea is its easy implementation.

Since the sequence of estimates produced by the OSL algorithms would achieve maximum penalized-likelihood, they also achieve minimum penalized-I-divergence as we discussed earlier. Thus, the next section derives algorithms to minimize the penalized I-divergence formulation given in Eq. (137) by adapting this OSL idea.

7.3.5 Application of Green's OSL

Applying Green's OSL algorithm, we obtain

$$\begin{aligned}
&D^{10}Q(\hat{a}^{(new)}|\hat{a}^{(old)}) - D\Phi(\hat{a}^{(old)};\alpha) \\
&= -\sum_j P(T_i,\nu_j) + \frac{\hat{a}^{(uc)}(T_i)}{\hat{a}^{(new)}(T_i)} - D\Phi(\hat{a}^{(old)};\alpha).
\end{aligned} \tag{155}$$

Setting Eq. (155) equal to zero suggests

$$\begin{aligned}
\hat{a}^{(new)}(T_i) &= \frac{1}{\phi(T_i) + D\Phi(\hat{a};\alpha)\big|_{\hat{a}=\hat{a}^{(old)}}}\ \hat{a}^{(uc)}(T_i) \\
&= \frac{\hat{a}^{(old)}(T_i)}{\phi(T_i) + D\Phi(\hat{a}^{(old)};\alpha)} \sum_j P(T_i,\nu_j)\frac{W(\nu_j)}{\hat{W}^{(old)}(\nu_j)},
\end{aligned} \tag{156}$$

where

$$\hat{W}^{(old)}(\nu_j) = \sum_i P(T_i,\nu_j)\hat{a}^{(old)}(T_i). \tag{157}$$

The same iteration was also found by Green for the Poisson data model [41, p.450]. Actually, our algorithm in Eq. (156) may be interpreted as an asymptotic version of the penalized EM algorithm for the Poisson data model [107].

7.3.5.1 Discussion on the Algorithms

One advantage of our algorithm in Eq. (156) is that it gives preliminary insights on how the penalties will effect the estimates. Consider the L_1-norm penalty. Then the derivative of this penalty is simply

$$D\Phi_{L_1}(\hat{a}^{(old)}; \alpha) = \alpha_i, \tag{158}$$

where α_i is the *i-th* component of α, namely the regularization parameter associated with the *i-th* component in the estimate $\hat{a}(T_i)$. Eq. (156) suggests that the algorithm simply encourages each component of the estimate to shrink at each iteration by a certain amount related to the regularization parameter.

Similar, but slightly more complicated, behavior can be inferred for the entropy-like penalty. Consider the derivative of the entropy-like penalty:

$$D\Phi_E(\hat{a}^{(old)}; \alpha) = \alpha_i(1 + \log \hat{a}^{(old)}(T_i)). \tag{159}$$

When this is embedded into the algorithm, the form in Eq. (156) suggests that if a specific component $\hat{a}(T_i)$ is excessively large, then a large number is added to the denominator in Eq. (156), resulting in a relatively big shrinkage of the component, compared with smaller components that will experience less shrinkage.

Desirable effects from the entropy-like penalty are best manifest when the regularization parameter vector α is chosen appropriately. If α is chosen too large compared to the normalization term ϕ, then the estimate will be overly shrunk. Therefore, the regularization should be customized carefully. Devising effective, general approaches for finding good regularization parameters is an active research field.

For Good's roughness penalty, the derivative of the penalty is given by

$$D\Phi_G(\hat{a}^{(old)};\alpha) = \alpha_i\left\{2\log\hat{a}^{(old)}(T_i) - \log\hat{a}^{(old)}(T_{i-1}) - \log\hat{a}^{(old)}(T_{i+1}) + 2\right\}$$
$$-\frac{\alpha_{i+1}\hat{a}^{(old)}(T_{i+1}) + \alpha_{i-1}\hat{a}^{(old)}(T_{i-1})}{\hat{a}^{(old)}(T_i)}. \tag{160}$$

Although it is difficult to interpret the effects of the roughness penalty based on this formula, similar issues concerning the choice of α_i arise.

7.4 Numerical Investigation

7.4.1 Experimental Settings

We focus on five different patterns to study the performance of our algorithms: a Gaussian-like function, a triangle, a double Gaussian-like function, a double triangle, and a rectangle. These patterns are generated as follows:

1) Gaussian-like function:

$$a(T_i) = \exp\left(-\frac{(T_i-400)^2}{4000}\right),\ 100 \le T_i \le 700, \tag{161}$$

2) Triangle:

$$a(T_i) = \begin{cases} 0, & 100 \le T_i < 300 \\ 1 - \frac{1}{100}|T_i - 400|, & 300 \le T_i \le 500 \\ 0, & 500 < T_i \le 700 \end{cases}, \tag{162}$$

3) Double-Gaussian-like function:

$$a(T_i) = \exp\left(-\frac{(T_i-300)^2}{1000}\right) + \exp\left(-\frac{(T_i-500)^2}{1000}\right),\ 100 \le T_i \le 700, \tag{163}$$

4) Double Triangle:

$$a(T_i) = \begin{cases} 0, & 100 \le T_i < 250 \\ 1 - \frac{1}{50}|T_i - 300|, & 250 \le T_i \le 350 \\ 0, & 350 < T_i < 450 \\ 1 - \frac{1}{50}|T_i - 500|, & 450 \le T_i \le 550 \\ 0, & 550 < T_i \le 700 \end{cases}, \tag{164}$$

5) Rectangle:

$$a(T_i) = \begin{cases} 0,\ 100 \leq T_i < 300 \\ 1,\ 300 \leq T_i \leq 500 \\ 0,\ 500 < T_i \leq 700 \end{cases} . \tag{165}$$

7.4.2 Reconstructions from Noiseless Measurements

7.4.2.1 Easy Patterns: Gaussian-like and Triangle Patterns

Figures 86(a) and 86(b) (via asterices) show the final estimates of the Gaussian-like and triangle patterns produced by our unconstrained algorithm when the measurements contain no noise.

For the Gaussian-like pattern, the estimate produced by our unconstrained algorithm given in Eq. (133) conforms nicely to the truth pattern. The estimate sequence $\hat{a}^{(k)}$ for this simple pattern converges much faster than the other patterns' estimate sequences (see Table 18 in Appendix C.1).

The unconstrained algorithm also produces an estimate that is reasonably close to the truth in the case of the triangle pattern. However, the algorithm converges extremely slowly in this case; the estimate is still evolving even at the 100-million-*th* iteration. (This excessive number of iterations was just performed for a thorough analysis; it would be unnecessary overkill for everyday use.)

7.4.2.2 Challenging Patterns: Double Gaussian-like and Double Triangle Patterns

Figures 86(c) and 86(d) show the final unconstrained estimates of the double-Gaussian-like and double-triangle patterns produced from noiseless measurements. The estimates for both cases are reasonably close to the truth patterns. Convergence is slow, as in the case of the triangle pattern (see Table 18 in Appendix C.1). An example of slow convergence of the unconstrained algorithm is shown in Fig. 87.

Note that in both cases, the estimates conform nicely with the corresponding truth patterns for temperatures above 400 K, but the estimates for temperatures below 400 K show a bit of disagreement. This is because the measurements contain more information

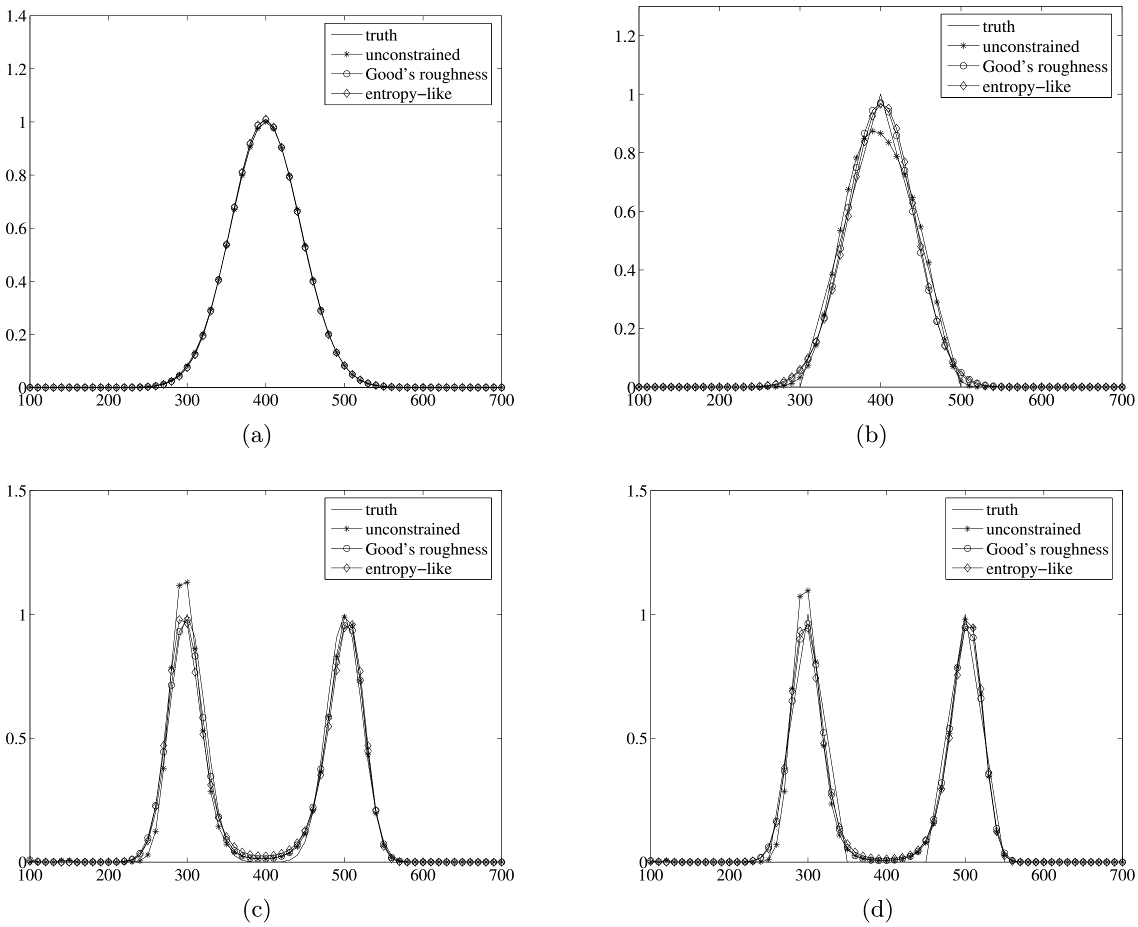


Figure 86: Estimates produced by the unconstrained and penalized minimum *I*-divergence algorithms from noiseless measurements for the (a) Gaussian-like, (b) triangle, (c) double Gaussian-like, and (d) double-triangle patterns. Each subfigure shows a truth pattern, an estimate when Good's roughness penalty is applied, and an estimate when our entropy-like penalty is applied.

about high-temperature portions than low-temperature portions. More precisely, the integral equation kernel illustrated in Fig. 88(a) puts much more weight on high-temperature portions than low-temperature portions. This unbalance in the amount of information results from properties of the integral equation kernel as discussed in the introduction. Note that there is about a 190 dB difference between two ends of the kernel in Fig. 88(a): its maximum is 4.3×10^{-9} and its minimum is 1.8×10^{-12}. Note that the kernel is a function of T_i and ν_j, but we take the summation of the kernel $\phi(T_i, \nu_j)$ over all ν_j for a fixed T_i to illustrate the limiting behavior discussed in the introduction. An example measurement is shown in Fig. 88(b).

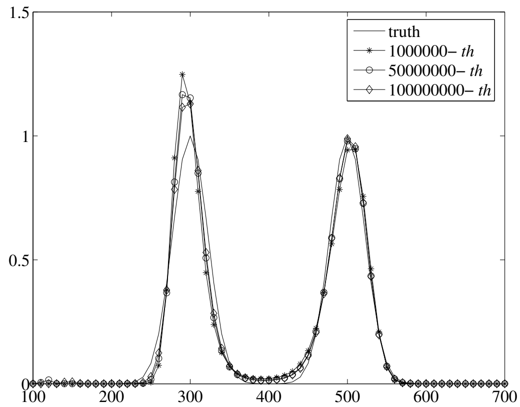
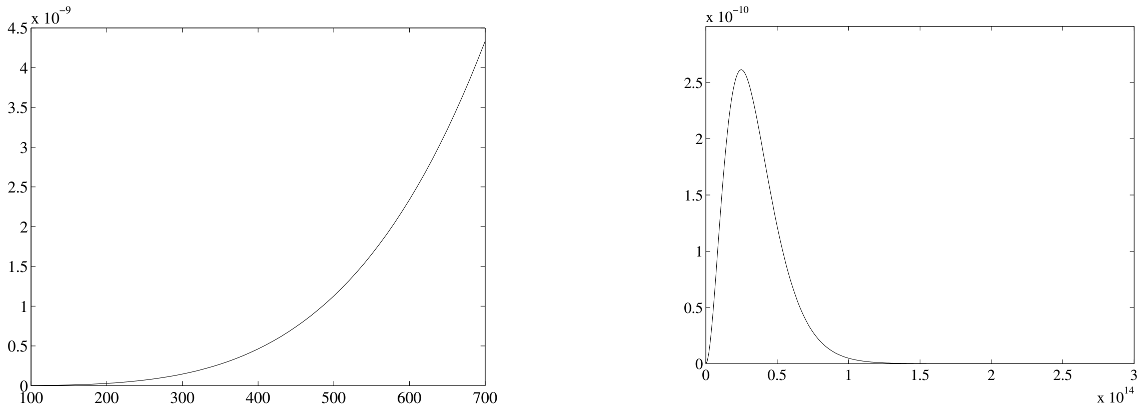


Figure 87: Example of slow convergence of the unconstrained algorithm. Some selected estimates are shown.



(a) (b)

Figure 88: (a) Visualization of the integral equation kernel ϕ; a summation was taken over all ν_j for a fixed T_i to best show the overall limiting behavior. (b) An example of a measurement W.

7.4.2.3 Edge Artifacts

Figure 89(a) shows a rectangle pattern and an estimate produced by our unconstrained algorithm from noiseless measurements. Since the pattern has discontinuities at 300 K and 500 K that cannot be perfectly approximated with bandlimited data, the algorithm tries to find an estimate closest to the true pattern in the sense of the *I*-divergence. This results in an *edge-artifact* phenomenon. It results from the ill-posedness of the problem, particularly the incompleteness of the kernel that must be truncated in practical implementation. Edge artifacts are well analyzed and discussed in [105] and [104].

Edge artifacts are usually manifest as overshoots near the locations where discontinuities exist and as ringing between discontinuities. Since the rectangle pattern in Fig. 89(a) does not have wide support, ringing does not show clearly, but the overshoots near the two discontinuities are quite clear. To better illustrate edge artifacts, we explore another rectangle pattern with wider support. Figure 89(b) shows this wider rectangle and the estimate produced by the unconstrained algorithm from noiseless measurements; here we observe both overshoot and ringing.

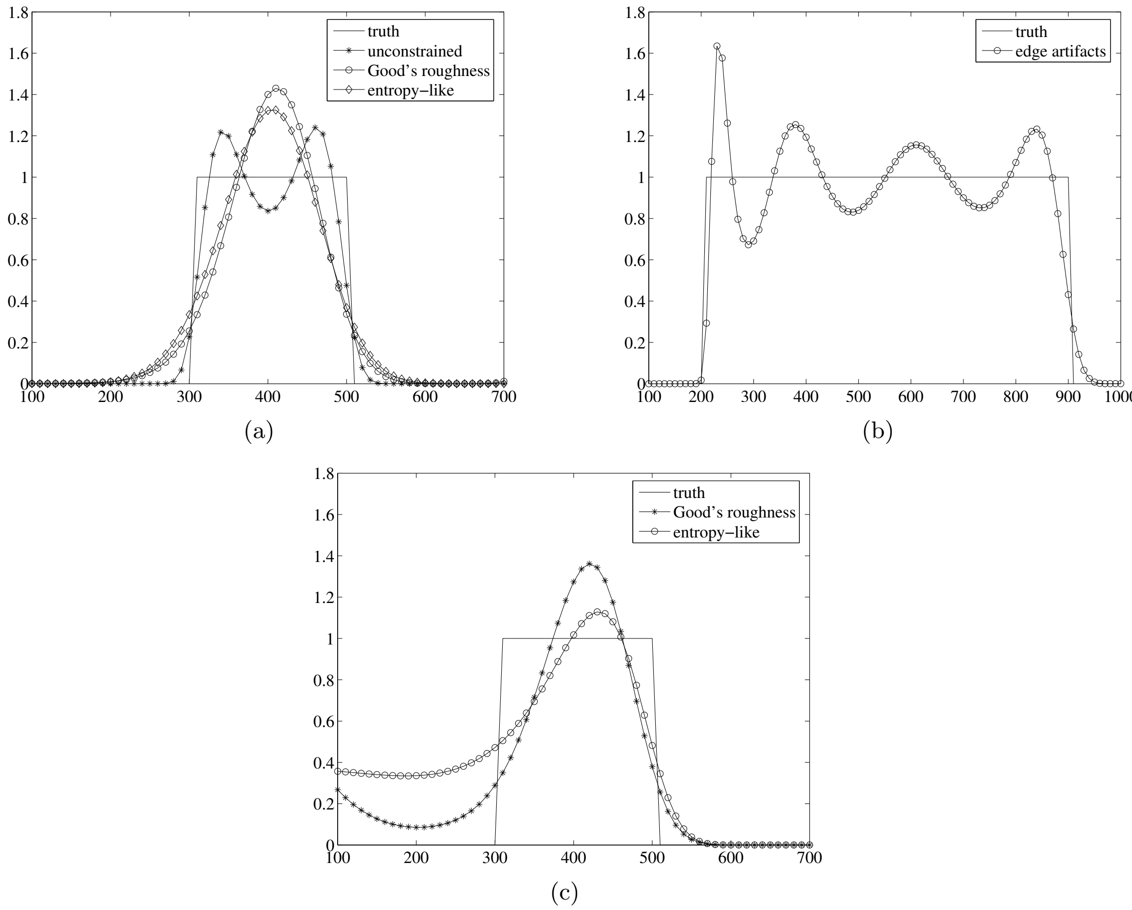


Figure 89: Final estimates of the rectangle pattern: (a) Rectangle estimates produced by the unconstrained algorithm, the constrained algorithm with Good's roughness penalty, and the constrained algorithm with our entropy-like penalty from noiseless measurements. The regularization parameter vector varies with temperature. (b) Rectangle estimates produced by the unconstrained algorithm from noiseless measurements. This shows the edge artifacts more clearly. (c) Estimates of the rectangle pattern used in Fig. 89(a), produced by the constrained algorithm given in Eq. (156), when the measurements are not corrupted by noise. In this case, the regularization parameters are constant with respect to temperature for both Good's roughness and our entropy-like penalties: 7×10^{-13} and 2×10^{-12}, respectively.

7.4.2.4 Regularization and Regularization Parameter Vectors

The clear manifestation of ill-posedness as edge artifacts motivated us to apply regularization methods to our minimum I-divergence framework. Figure 89(c) shows the regularized (or constrained) estimates produced by our regularized minimum I-divergence algorithms, given in Eq. (156), from noiseless measurements. Good's roughness and entropy-like functions are used to regularize the estimates, and constant regularization parameter vectors are applied as in Eq. (136).

An interesting situation occurs when a constant regularization parameter vector is used. Through trial and error, we found that the estimates shown in Fig. 89(c) are approximately "best" in the sense that bigger parameters would result in numerical errors, and smaller parameters would not provide enough smoothing to suppress the edge artifacts. However, these "best" estimates appear to be "over-regularized" for temperatures below 400 K. The estimate produced via entropy-like regularization has a left-side tail that drags on undesirably long and seems enchained toward a certain value. Recall that pure entropy-like penalties shrink estimates toward $1/e$ (0.3679) unless the algorithm leads the estimates toward some other value dependent upon the observed data. Note which value the estimate produced with the entropy-like penalty in Fig. 89(c) seems inclined towards. This suggests that a constant parameter vector would be too large for temperatures below 400 K if the constant is chosen such that it is large enough to provide the estimate in the temperature range above 400 K with an appropriate amount of smoothing. Consequently, the effect of the entropy-like penalty becomes dominant over the effect of the data through the I-divergence term for low temperatures, and thus the estimate values in the temperature below 400 K have a tendency toward $1/e$. Similar arguments may be drawn from the estimates produced with Good's roughness penalty.

Numerical errors occurring when a constant parameter vector is used may be understood best in view of the iteration in Eq. (156). Recall that the minimum of ϕ is much smaller than the maximum of ϕ; there is about 190 dB difference between the two values. Note that the derivative of the penalty could become negative. Therefore, if the (constant) regularization parameter is relatively too large with respect to ϕ for some temperatures T_i, then the denominator in Eq. (156) could go negative, violating the nonnegativity constraints on the estimate and spoiling the entire estimate at the next iteration.

To avoid such numerical catastrophes and the issue of over-regularization for low temperatures, we design a regularization parameter vector that varies with T_i. Concerning the numerical errors, a useful choice would be a regularization parameter function that depends on ϕ: $c\phi(T_i)$, where c is a customizable constant. This helps us avoid numerical troubles. However, with this choice of parameter, we still often have the problem of over-regularization (or under-regularization).

Since one end of the parameter function always determines the other end when using the regularization parameter function $c\phi(T_i)$, we often observe the over-regularization or under-regularization around the two ends of the estimates. To compensate for this, we allow further flexibility in the parameter function by setting the two ends of the parameter function as customizable variables, while still incorporating the shape of the kernel into the parameter function. More specifically, we suggest the following parameter function for regularization:

$$\alpha(T_i) = \frac{\phi(T_i) - \max_i(\phi(T_i))\mathbf{1}}{\max_i(\phi(T_i)) - \min_i(\phi(T_i))}(c_{max} - c_{min}) + c_{min}\mathbf{1}, \tag{166}$$

where $\mathbf{1}$ denotes a vector whose components are all 1 and whose length is the same as that of ϕ. The two constants c_{max} and c_{min} are customizable and determine the two ends of the parameter function, whose shape is similar to the kernel.

Figure 89(a) shows final estimates produced by our constrained algorithms with Good's roughness and the entropy-like penalties when we apply the parameter function specified in Eq. (166). The iteration shown and the exact values of c_{max} and c_{min} are given in Tables 19 and 20 in Appendix C.1. The algorithm successfully avoids the problem of negative estimates and over-regularization and produces reasonably good estimates.

Another merit of applying regularization via a penalty is that the iterations converge much faster than with the unconstrained algorithm. Interestingly, this is consistent with what has been observed in emission tomography (see [31] and its references).

7.4.2.5 *Short Note on the L_1-norm Penalty*

We also applied the L_1-norm penalty for regularization. However, unlike the maximum entropy-like penalty, the L_1-norm only had a trivial effect on the estimates: the estimate is simply shrunken toward zero while shapes of estimates are not noticeably changed. On one hand, this is consistent with our interpretation of the algorithm given in Eq. (156) involving the L_1-norm-penalty derivative given in Eq. (158). On the other hand, it is surprising and counterintuitive that the L_1-norm and entropy-like penalties result in totally different effects on the estimates even though they operate on a similar principle in the sense that both encourage shrinkage of estimates.

Pertaining to this discussion, it is also interesting and surprising that Good's roughness and the entropy-like penalties have very similar impact on the estimates even though they operate on entirely different principles. Recall that Good's roughness provides smoothing of estimates based on the spatial relationship between neighboring components, while the entropy-like function works on each component of an estimate independently of the other components of the estimate. Some discussion of this unusual behavior is provided in [26].

7.4.2.6 *Characteristics of Ill-posedness*

Note that the overshoots in the two unconstrained estimates shown in Figs. 89(a) and 89(b) have different heights. We have discussed how the high-temperature portions of $a(T)$ contribute to the measurements more than the low-temperature portions. We may readily infer that such an unbalanced contribution would become more serious when the support of the truth pattern becomes larger. Furthermore, this argument indicates that the support of an estimate may change the degree of ill-posedness. The two different sets of two overshoots in Figs. 89(a) and 89(b) support this observation.

7.4.2.7 *Regularized Reconstructions from Noiseless Measurements*

Figure 86 shows final estimates produced by our regularized minimum I-divergence algorithms when Good's roughness and entropy-like penalties are incorporated. The parameter function is designed by Eq. (166); the control variables c_{max} and c_{min} were determined by trial and error. (It is generally not a trivial issue to determine appropriate parameter values for regularization.) The convergence speeds and parameter control values are given in Appendix C.1.

Both of the penalties with the proposed parameter function produce very nice estimates for all four patterns. All the estimates quite precisely agree with the corresponding truth patterns. Particularly, the penalties successfully regularize the low temperature portions of the estimates and produce estimates that match the associated truth patterns well; the little overshoots produced by the unconstrained algorithm in both Figs. 86(c) and 86(d) are properly suppressed.

As in the reconstruction of the rectangle pattern, both Good's roughness penalty and our entropy-like penalty improve the convergence speed significantly. In general, the estimates produced with Good's roughness penalty converge a little faster than those produced with the entropy-like penalty. For details, compare the convergence iterations in Appendix C.1.

7.4.3 Reconstructions from Noisy Measurements

To realize noisy measurements, we add uniformly distributed random noise as follows:

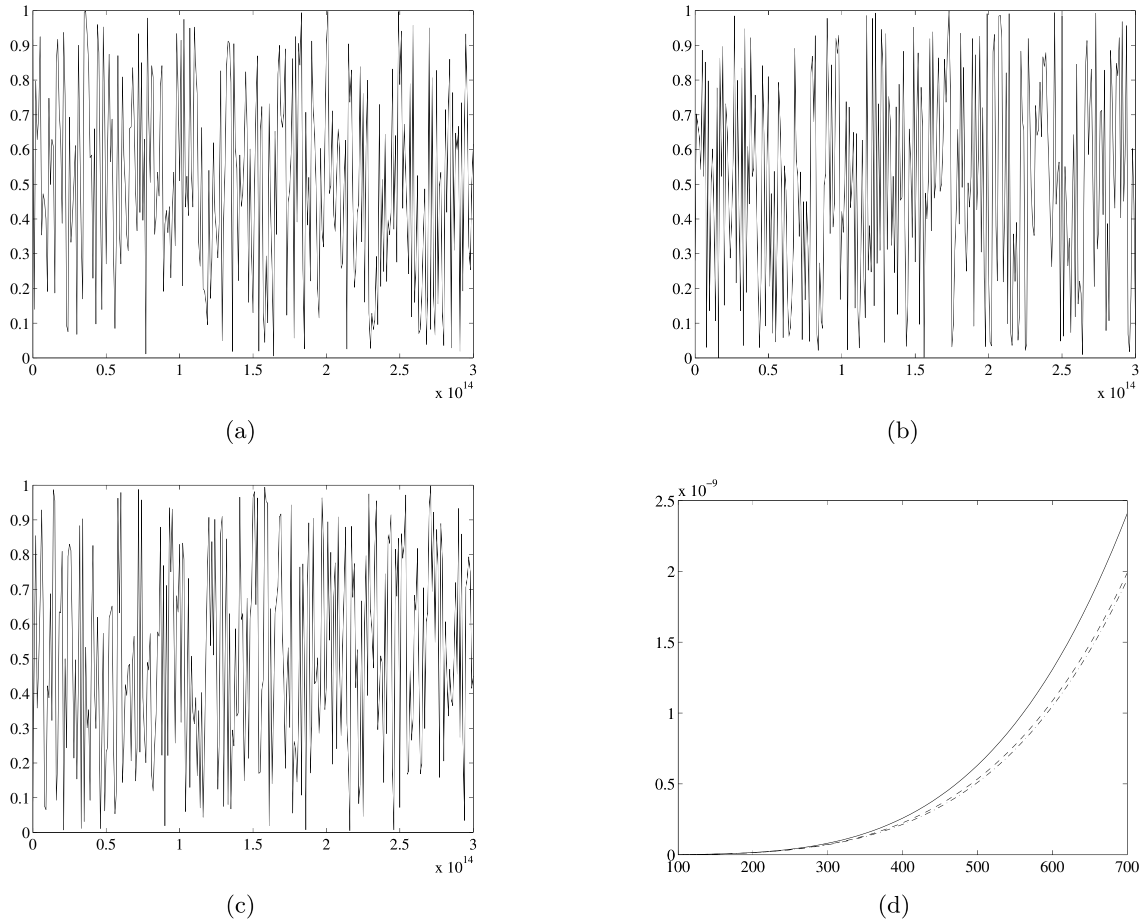
$$W(\nu_j) = \max\left(W_{un}(\nu_j) + k_n N(\nu_j), 0\right), \tag{167}$$

where k_n represents a noise level control variable, which is either 10^{-12} or 10^{-13}, and N denotes a random noise vector that is uniformly distributed over [-0.5 0.5] and has the same length as that of the measurements W_{un}, which denotes noiseless measurements generated purely by the integral equation system given in Eq. (122) for a truth pattern. Note that since the I-divergence is defined only on nonnegative functions, the noisy measurements that become negative are forced to zero.

7.4.3.1 Noise Back Propagation

To investigate how noise may affect the estimates through the algorithms, consider three different random noise vectors, with components independently uniformly distributed over [0 1], whose lengths are the same as the length of the measurements. Figures 90(a) through

90(c) show three different uniform random noise realizations. Note that noise that is superimposed upon the measurements W is back-propagated by the kernel as indicated by the term $\hat{W}^{(k)}$ in Eq. (156). Figure 90(d) illustrates how noise can have an impact on estimates when they are back-propagated by the kernel ϕ. Noticeably, three completely different noise realizations back-propagate to similar shapes. The figure shows that more noise is back-propagated to the high temperature portion than the low temperature portion. The shape of the back-propagated noise is similar to the kernel. This heuristically justifies our use of the parameter function specified in Eq. (166).



(a) (b) (c) (d)

Figure 90: (a)-(c) Three different realizations of uniformly distributed random noise. (d) Results of noise realizations in Figs. 90(a) through 90(c) when back-propagated by the integral equation kernel ϕ.

7.4.3.2 Noise Artifacts: Unconstrained Reconstructions

Figure 91 shows estimates produced by the unconstrained minimum I-divergence algorithm. Noise entirely confuses the algorithm and spoils the estimates. For some truth patterns such as the rectangle pattern, the algorithm even tries to reconstruct some radiances at temperatures where it should not be (see the signals indicated via x-marks between 100 K and 200K in the estimates of the rectangle in Fig. 91(b)). Similar effects are observed in the estimates of the Gaussian and the triangle in Figs. 91(a) and 91(c), respectively.

Another common effect is that the estimates become very narrow and large compared with the true area temperature distribution. This effect motivated regularization by the entropy-like penalty.

Also note that noise confuses the algorithm such that it cannot accurately locate bumps in the area temperature distribution. Even though the degree of confusion varies from pattern to pattern, we can clearly observe this effect in the reconstructions of the double-Gaussian-like and the double-triangle patterns. Note how much shifted to the left the two bumps that are located below 400 K are.

7.4.3.3 Regularized Reconstructions from Noisy Measurements

Figures 92 through 96 show estimates produced by our regularized minimum I-divergence algorithms with Good's roughness and entropy-like penalties. The regularization parameter functions are set as in Eq. (166).

For comparison, we show two estimates for a fixed noise level k_n and a penalty. More specifically, when the noise level is low ($k_n = 10^{-13}$), the estimates when α is low (circles) represent appropriately regularized solutions, and the estimates when α is high (x-marks or diamonds) represent overly regularized solutions. When the noise level is high ($k_n = 10^{-12}$), the estimates with a high α represent appropriately regularized solutions, and the estimates with a low α represent under-regularized solutions, meaning the parameter function is set such that the penalty does not provide enough smoothing to the estimates.

The algorithms succeed in reconstructing reasonably good shapes for all the patterns, especially when the noise level is low. In particular, the estimates for the Gaussian-like, the rectangle, and the triangle patterns conform to their corresponding truth patterns quite well. Even when the measurement noise level is high, the algorithms successfully reconstruct nice, smooth estimates of the rectangle pattern without edge artifacts, even though the signal maximum becomes higher than the estimates from noiseless measurements.

An interesting effect of over-regularization may be observed in Figs. 92(a) and 92(b). When the estimates are overly regularized, the estimates are pulled down and spread. However, the spread occurs mainly toward the left of the estimates, with both penalties. Similarly, when the estimates are under-regularized, the estimates become higher and shrinkage behavior occurs mainly at lower temperatures.

It would be inevitable that sufficiently large noise in the minimum I-divergence framework destroys information on bump location. Recall that the unconstrained algorithm is confused by noise and produces estimates shifted toward the left, especially when noise level is high (as seen in Fig. 91). Regularization cannot help the algorithms avoid these intriguing artifacts. Observe the estimates of the double-Gaussian-like and the double-triangle patterns shown in Figs. 95 and 96, which show such artifacts most clearly. Even though the shapes and magnitudes are successfully reconstructed, the two bumps in the double-Gaussian-like pattern and the two triangles in the double-triangle pattern are apparently shifted to the left, even with the regularization. Note that the locations of the bumps and triangles are consistent with the locations of the spikes reconstructed by the unconstrained algorithm from noisy measurements. The other three patterns, the Gaussian-like, the triangle, and the rectangle, show similar artifacts as well, but less severely.

7.5 Conclusions and Future Work

We developed an iterative algorithm that attempts to find the area temperature distribution for the inverse blackbody radiation problem based upon an information-theoretic discrepancy measure called Csiszár's I-divergence.

When measurements are not corrupted by noise, our unconstrained algorithm produces reasonably good estimates. However, in practice, measurements always contain noise. Once noise is involved, the unconstrained algorithm can no longer produce reasonable estimates.

We proposed methods of regularization to overcome this problem. Good's roughness and entropy-like penalties were suggested to suppress the observed edge and noise artifacts and succeeded in producing "reasonable" estimates.

Even though regularization can lead to reasonable solutions, there will always be some information that cannot be recovered. In particular, when the noise level is high, algorithms may become confused and misinterpret the locations of bumps; this information cannot be restored by typical regularization techniques.

To solve the penalized-I-divergence optimization problem, we used the theoretical aspect that minimizing a penalized-I-divergence is equivalent to maximizing the corresponding penalized-likelihood. Since a certain type of EM algorithm can achieve maximization of the penalized-likelihood, it can also minimize the penalized-I-divergence. This motivated us to apply Green's OSL algorithms, originally designed for penalized EM algorithms, to achieve minimization of the penalized-I-divergence.

Penalized EM algorithms are methods for finding maximum penalized-likelihood estimates, but they often converge slowly. Space Alternating Generalized EM (SAGE) algorithms are techniques invented by Fessler and Hero [31] for accelerating penalized EM algorithms. Therefore, in the future, we may be able to apply the SAGE idea to our I-divergence framework to improve the convergence speed of our minimum penalized-I-divergence algorithms.

This section has focused entirely on regularization via penalty methods. Another option might be to use Grenander's method of sieves, or resolution kernels, or a combination of the sieves and kernels, as described on pp. 147–165 of [104]. This could be slightly tricky due to the spatially-varying nature of the integral kernel in Planck's law.

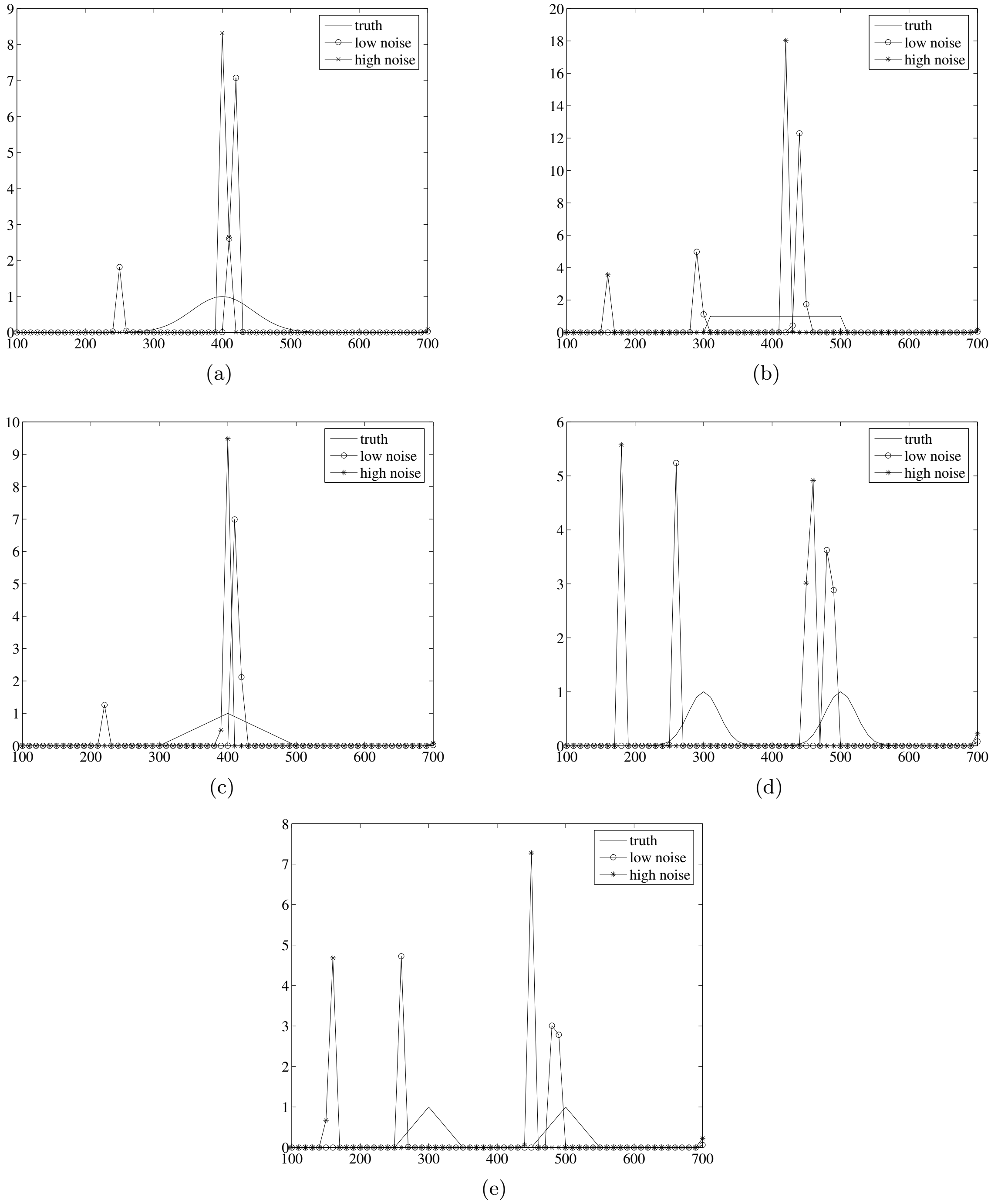


Figure 91: Final estimates produced by our unconstrained minimum I-divergence algorithm from noisy measurements for the (a) Gaussian-like, (b) rectangle, (c) triangle, (d) double-Gaussian-like, (e) double-triangle patterns.

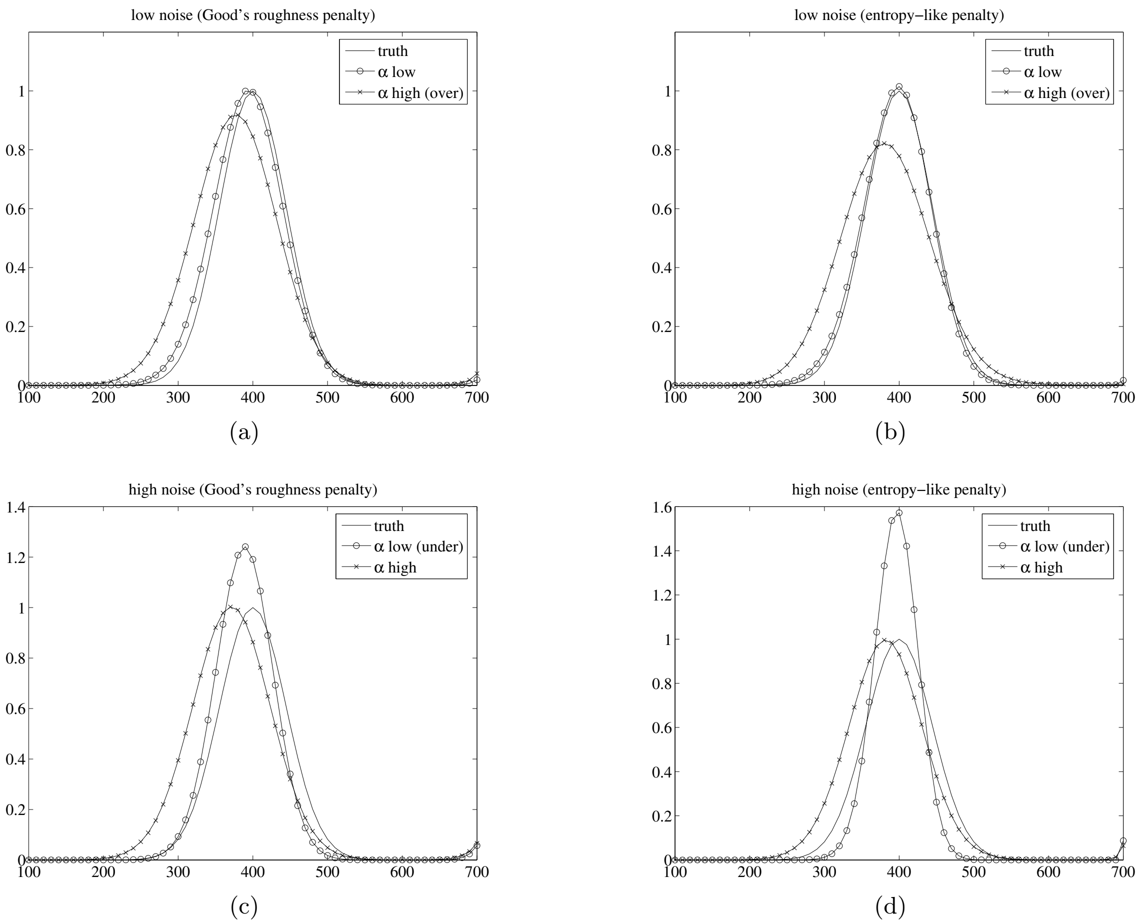


Figure 92: Final estimates produced by the regularized minimum I-divergence methods from noisy measurements for the Gaussian-like pattern when (a) the noise level k_n is 10^{-13}, and Good's roughness penalty is applied, (b) the noise level k_n is 10^{-13}, and our entropy-like penalty is applied, (c) the noise level k_n is 10^{-12}, and Good's roughness penalty is applied, (d) the noise level k_n is 10^{-12}, and our entropy-like penalty is applied. Each subfigure shows estimates produced with low α and high α.

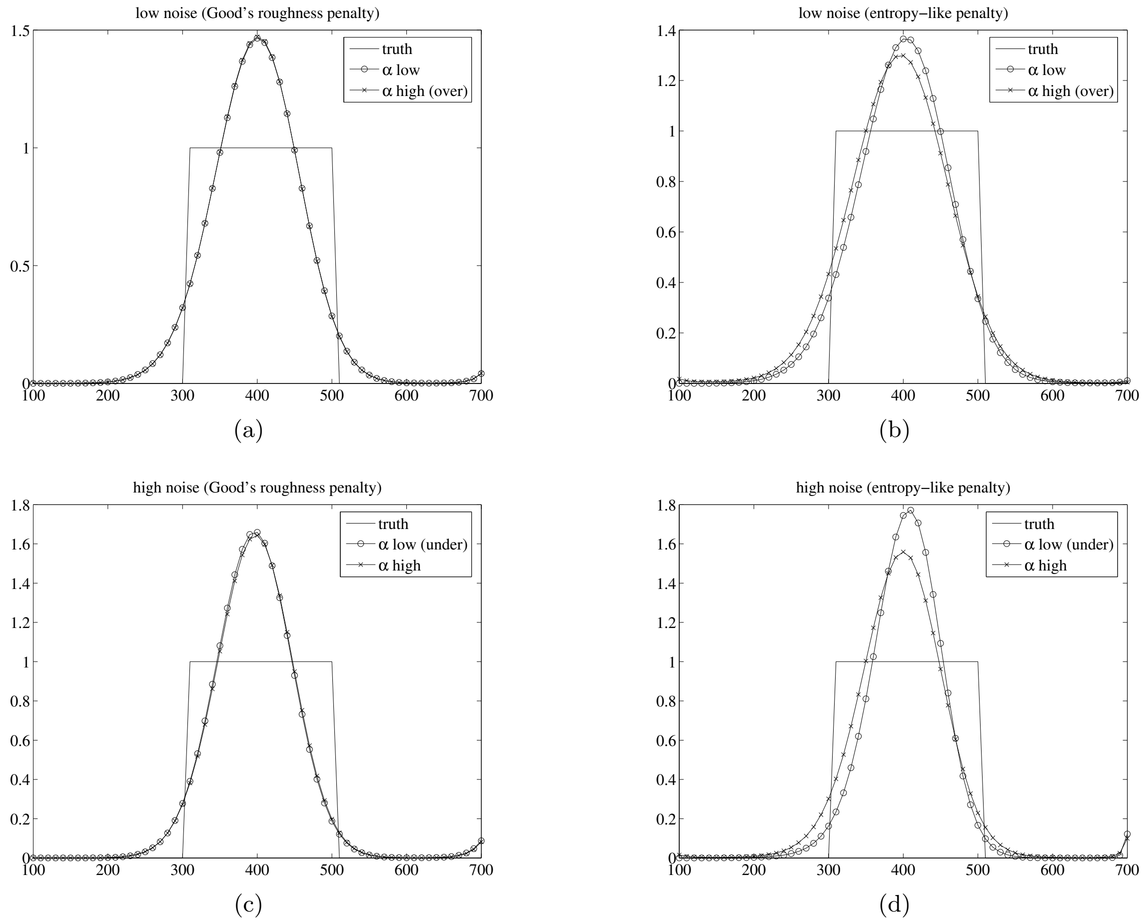


Figure 93: Final estimates produced by the regularized minimum I-divergence methods from noisy measurements for the rectangle pattern when (a) the noise level k_n is 10^{-13}, and Good's roughness penalty is applied, (b) the noise level k_n is 10^{-13}, and our entropy-like penalty is applied, (c) the noise level k_n is 10^{-12}, and Good's roughness penalty is applied, (d) the noise level k_n is 10^{-12}, and our entropy-like penalty is applied. Each subfigure shows estimates produced with low α and high α.

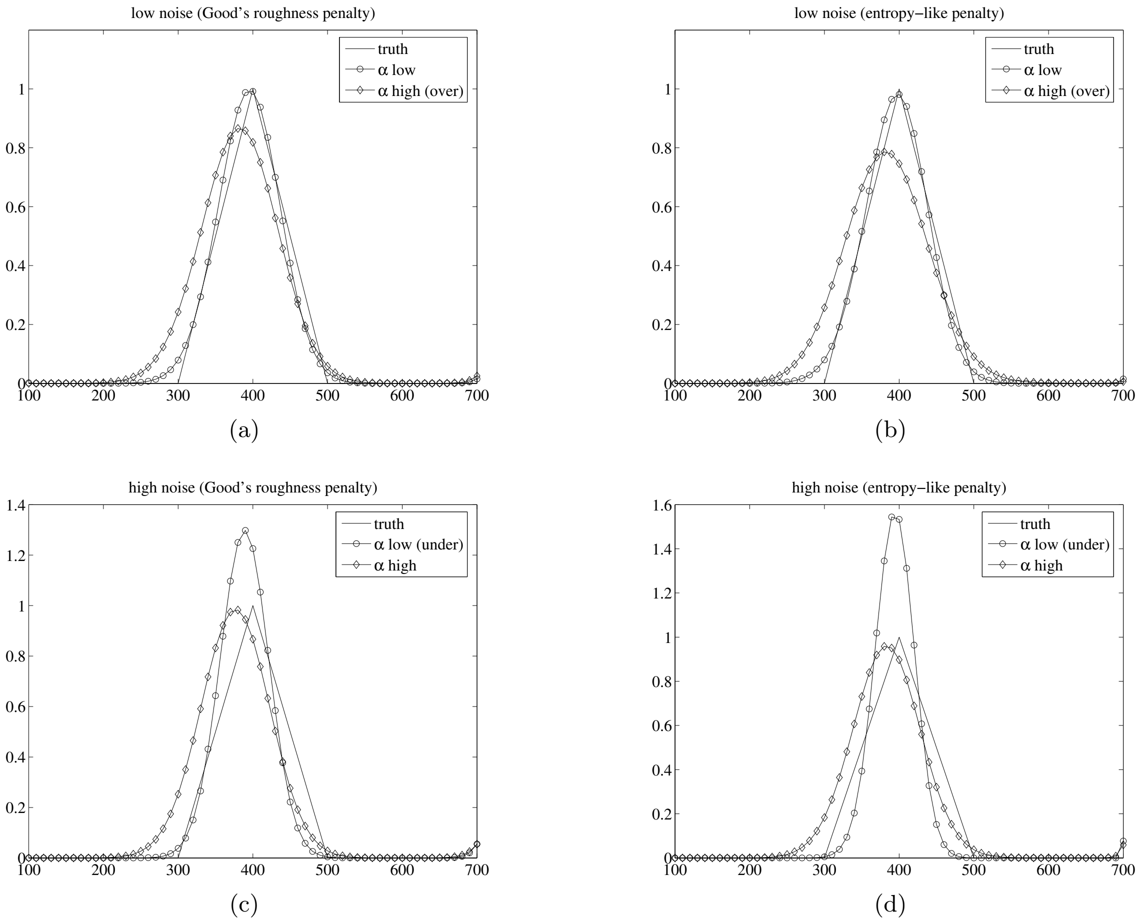


Figure 94: Final estimates produced by the regularized minimum I-divergence methods from noisy measurements for the triangle pattern when (a) the noise level k_n is 10^{-13}, and Good's roughness penalty is applied, (b) the noise level k_n is 10^{-13}, and our entropy-like penalty is applied, (c) the noise level k_n is 10^{-12}, and Good's roughness penalty is applied, (d) the noise level k_n is 10^{-12}, and our entropy-like penalty is applied. Each subfigure shows estimates produced with low α and high α.

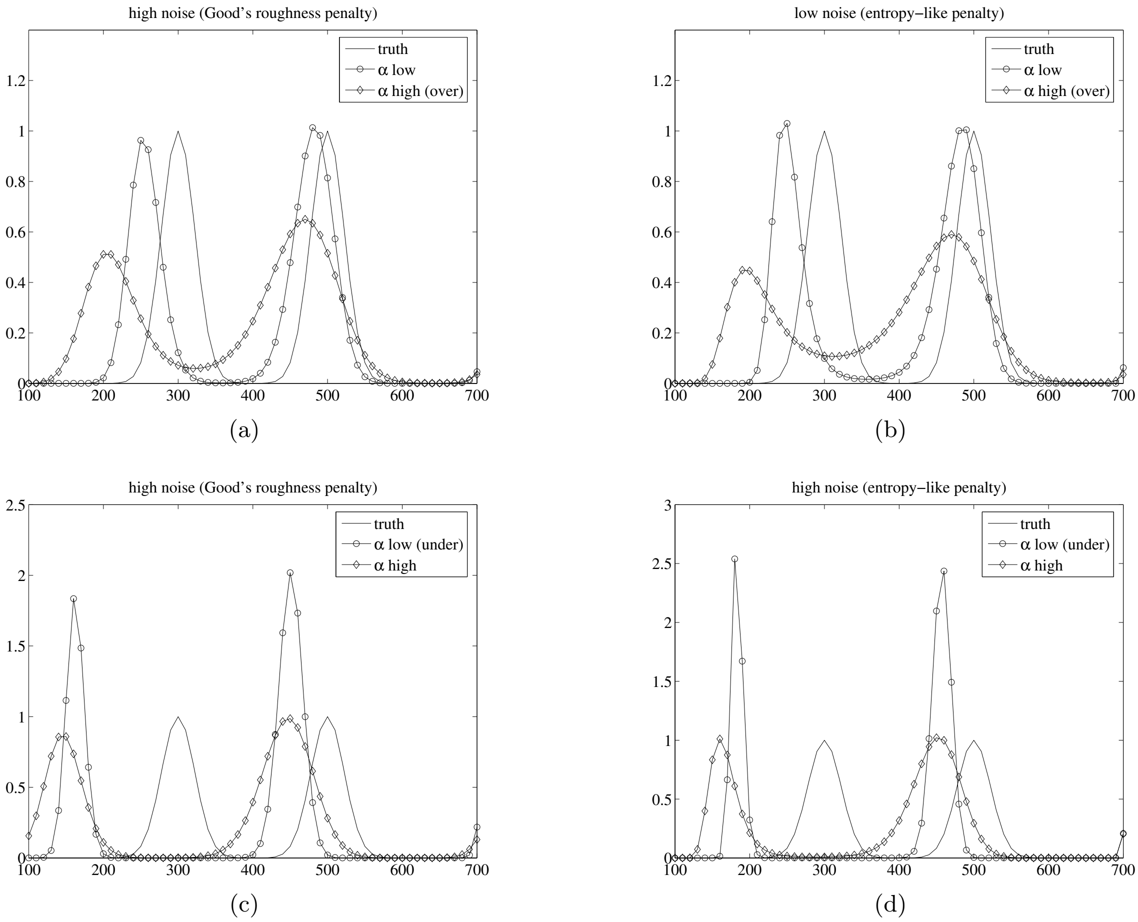


Figure 95: Final estimates produced by the regularized minimum I-divergence methods from noisy measurements for the double-Gaussian-like pattern when (a) the noise level k_n is 10^{-13}, and Good's roughness penalty is applied, (b) the noise level k_n is 10^{-13}, and our entropy-like penalty is applied, (c) the noise level k_n is 10^{-12}, and Good's roughness penalty is applied, (d) the noise level k_n is 10^{-12}, and our entropy-like penalty is applied. Each subfigure shows estimates produced with low α and high α.

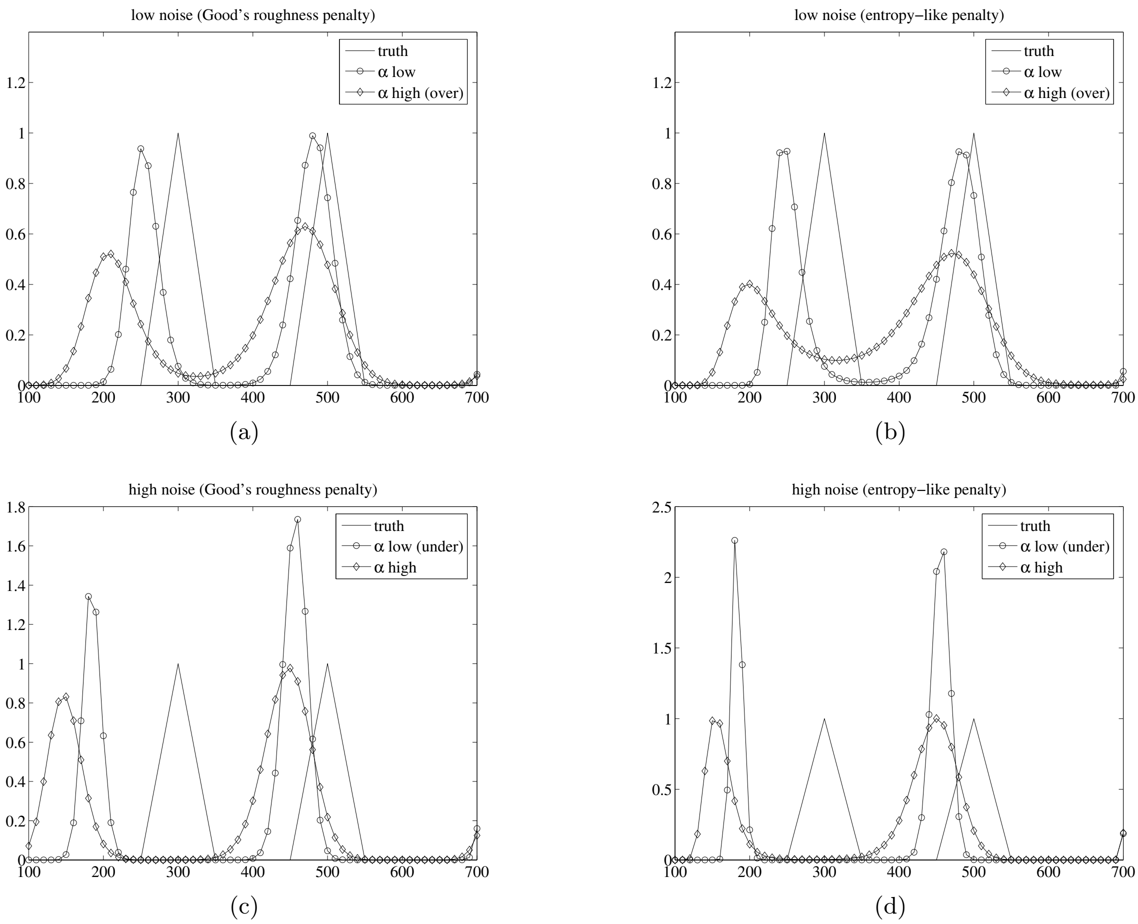


Figure 96: Final estimates produced by the regularized minimum I-divergence methods from noisy measurements for the double-triangle pattern when (a) noise level k_n is 10^{-13}, and Good's roughness penalty is applied, (b) the noise level k_n is 10^{-13}, and our entropy-like penalty is applied, (c) the noise level k_n is 10^{-12}, and Good's roughness penalty is applied, (d) the noise level k_n is 10^{-12}, and our entropy-like penalty is applied. Each subfigure shows estimates produced with low α and high α.

CHAPTER VIII

CHANNEL INPUT DISTRIBUTION ESTIMATION USING MINIMUM I-DIVERGENCE ALGORITHM

8.1 Introduction

Fozunbal, McLaughlin, and Schafer [37] have recently presented results concerning the capacity of Rician channels. In their exposition, an integral equation is presented relating the input and the output distributions of Rician channels. Researchers exploring the Rician channel may propose input distributions and find the resulting output distribution. The inverse problem of finding the input distribution that yields, as closely as possible, a desired output distribution is much more difficult. To provide an analytic tool for researchers, we formulate an iterative algorithm for solving this inverse problem. Although this work was motivated by recent results on Rician channels, our algorithm could be applied to other channels as well.

8.1.1 Nonnegative Linear Inverse Problems

Problems involving the reconstruction of an input from a blurred output under a linear blurring function are omnipresent in engineering and science. In particular, inverse problems of linear systems with nonnegative parameters, subject to nonnegativity constraints on the solution, are often of interest. Vardi and Lee [114] showed that deterministic linear inverse problems with nonnegativity constraints can be thought of as statistical estimation problems from incomplete data based on an infinite number of observed samples, which allows us to use the weak law of large numbers. Hence, they showed that maximum-likelihood estimation and the EM algorithm provide a direct method of addressing such problems. Snyder *et al.* [107] address the same issue, and conclude that solutions obtained by minimizing Csiszár's I-divergence measure asymptotically correspond to certain maximum-likelihood estimators. They also showed that the sequence of estimators from their method has a nice set of properties such as guaranteed convergence to the global minimum, preserved nonnegativity of solutions, and monotonically decreasing I-divergence. Csiszár justified the use of his I-divergence measure by proving that, if all the functions involved are required to be nonnegative, minimizing his measure is the only choice consistent with the axioms he proposed. The algorithm proposed by Snyder *et al.* has been employed in various fields such as medical imaging, astronomical imaging, and image restoration [72], [88], [105].

Because all involved functions (the input, the output, and the kernel) in the problem of interest in this chapter are nonnegative, we apply the minimum I-divergence method. For Rician channels, Fozunbal *et al.* [37] have shown that the input distribution should be symmetric; hence, we derive a new algorithm that preserves the symmetry of the solutions. We also show that if the transition kernel is symmetric with respect to the origin (as is the case with Rician channels), then the new symmetry-preserving algorithm and the original algorithm produce the same estimate at each iteration, assuming the algorithms are initialized with the same *symmetric* initial estimate. Hence, although the proposed algorithms do not improve the rate of convergence per iteration or the quality of the solutions in the Rician case, they noticeably improve computation time.

We illustrate the algorithm with two kinds of scenarios. In the first scenario, we test the algorithm with a known input distribution to verify the accuracy of the algorithms. The second scenario matches how we expect researchers to use the algorithm, in that we give an output distribution to the algorithm and ask it to find the input distribution that gets as close as possible to the desired output, with the understanding that an input distribution that gives the *exact* desired output may not exist.

The estimated inputs may show some artifacts in some situations. These artifacts are also discussed.

8.1.2 The Channel Mapping

Let X and Y be random variables that represent the input and the output of a channel, respectively, defined over the entire real line. Consider a discrete-time channel specified by

$$Y = HX + N, \tag{168}$$

where H is a normal random variable with mean $\overline{h}$ and variance σ_h^2, and N is zero-mean additive white Gaussian noise with variance σ_n^2.

Let $F_X(x)$ and $F_Y(y)$ denote the distribution functions of X and Y, respectively. They are related as follows:

$$F_Y(y) = \int_{-\infty}^{y} \int_{-\infty}^{\infty} p(v|x) dF_X(x) dv, \tag{169}$$

where the kernel $p(v|x)$, the channel transition density [37], is specified by

$$p(v|x) = \frac{1}{\sqrt{2\pi(\sigma_h^2 x^2 + \sigma_n^2)}} e^{-\frac{1}{2(\sigma_h^2 x^2 + \sigma_n^2)}(v - \overline{h}x)^2}. \tag{170}$$

This chapter considers the case where the kernel $p(v|x)$ and the output density $p_Y(y)$ are known. We are concerned with estimating the input density $p_X(x)$. To estimate $p_X(x)$, we suggest using the minimum *I*-divergence algorithm proposed in [107].

8.1.3 Organization

This chapter is structured as follows. In Section 8.2, our application of the minimum *I*-divergence method is described, and the symmetry-preserving algorithms are proposed. Additionally, the equivalence of the estimates from the original algorithm and the proposed symmetry-preserving algorithms is proved. Simulation results are presented and analyzed in Section 8.3. Our discussion concludes in Section 8.4.

8.2 Algorithms

8.2.1 Minimum I-divergence Algorithm

Csiszár's *I*-divergence is an information-theoretic discrepancy measure between two nonnegative functions. This measure is a generalization of the Kullback-Leibler distance designed to consider functions whose integrals may not be equal. In [23], Csiszár concludes that if both functions being compared are required to be nonnegative, his *I*-divergence measure is the only discrepancy measure consistent with the axioms he proposes.

The authors of [107] proposed an algorithm for nonnegative linear inverse problems that minimizes this discrepancy measure. The algorithm produces a sequence of estimates with nice properties, such as guaranteed nonnegativity of every estimate in the sequence, monotone convergence to a global minimum, and so on.

The relation between the input and the output distributions in (169) is equivalent to a Fredholm equation of the first kind relating densities $f_Y(y)$ and $f_X(x)$:

$$f_Y(y) = \int_{-\infty}^{\infty} p(y|x) f_X(x) dx. \tag{171}$$

Note that in (171), all functions involved are nonnegative. This motivates applying the minimum I-divergence algorithm. For computer implementation, we assume the random variables X and Y are defined over finite dimensional sets $\mathcal{X} \subset \mathbb{R}$ and $\mathcal{Y} \subset \mathbb{R}$, respectively. Then (171) becomes

$$p_Y(y) = \sum_{x \in \mathcal{X}} p(y|x) p_X(x), \tag{172}$$

where $p_X(x)$ and $p_Y(y)$ are the probability mass functions of X and Y, respectively, and $p(y|x)$ has been similarly discretized. Since the discretization of densities of probability mass functions is just an artifice of computer implementation, and the underlying continuous densities are what we are really interested in, the remainder of this chapter will use the shorter term "density" instead of "probability mass function."[1]

Our goal is to find $\hat{p}_X(x)$, an estimate of $p_X(x)$ that minimizes Csiszár's I-divergence,

$$\begin{aligned} &I[p_Y(y)||\hat{p}_Y(y)] \\ &= \sum_y p_Y(y) \ln \left[\frac{p_Y(y)}{\hat{p}_Y(y)}\right] - \sum_y [p_Y(y) - \hat{p}_Y(y)], \end{aligned} \tag{173}$$

where $p_Y(y)$ is the given channel output, and the output $\hat{p}_Y(y)$ is generated by a particular estimate $\hat{p}_X(x)$ according to

$$\hat{p}_Y(y) = \sum_{x \in \mathcal{X}} p(y|x) \hat{p}_X(x). \tag{174}$$

Using the Kuhn-Tucker conditions, we can obtain an iterative algorithm [107] that minimizes

[1]Mathematically inclined readers that might be bothered by this abuse of termonology may think of these densities as Radon-Nikodym derivatives with respect to a discrete counting measure instead of the usual Lebegue measure.

(173), which is given by

$$\hat{p}_X^{(k+1)}(x) = \frac{\hat{p}_X^{(k)}(x)}{\sum\limits_{y\in\mathcal{Y}} p(y|x)} \sum_{y\in\mathcal{Y}} \left[\frac{p(y|x)p_Y(y)}{\sum\limits_{x'\in\mathcal{X}} p(y|x')\hat{p}_X^{(k)}(x')} \right]. \tag{175}$$

In the particular case of interest here, the summation of $p(y|x)$ in the denominator over all y given a fixed x is 1 because $p(y|x)$ is a probability mass function. However, the term is left in the expression because modifications to it will be made in a subsequent algorithm.

As mentioned before, a sequence produced by this algorithm has a nice set of properties (Section III of [107]). This kind of algorithm has found application in diverse areas. For example, Lucy [72] and Richardson [88] first derived it for image restoration problems in the 1970's using heuristic arguments.

The steps of the algorithm are described as follows:

1. Begin with an input estimate $\hat{p}_X^{(0)}(x)$ that is a valid probability mass function (non-negative).

2. Divide the known output density by the output density $\hat{p}_Y^{(k)}(y)$ derived by plugging $\hat{p}_X^{(k)}(x)$ into (174). Call this function $U^{(k)}(y)$.

3. Compute the summation over y in the numerator.
   Call this $W^{(k)}(x) = \sum_y p(y|x)U^{(k)}(y)$.

4. Update the estimate of $\hat{p}_X(x)$ by

$$\hat{p}_X^{(k+1)}(x) = \hat{p}_X^{(k)}(x)W^{(k)}(x). \tag{176}$$

5. Repeat steps 2 through 4 until a convergence criterion is met.

8.2.2 Symmetry-Preserving Minimum I-divergence Algorithm

8.2.2.1 Symmetry of the Input

Theorem 1 of [37] shows that the channels described in Section 8.1 have a bijection property: for a given symmetric output distribution, there exists a unique, symmetric input distribution that induces the given output distribution. Considering this property, we propose a modification of the algorithm given in Section 8.2.1 that preserves the symmetry of the estimated input density. The modified algorithm also exploits symmetry for computational efficiency.

Assume that the input density is known to be symmetric with respect to the origin. Given this assumption, we note the following relations:

$$\begin{aligned}
&I\left[p_Y(y)||\hat{p}_Y(y)\right] \\
&= I\left[p_Y(y)\middle|\middle|\sum_{x\in\mathcal{X}} p(y|x)\hat{p}_X(x)\right] \\
&= I\left[p_Y(y)\middle|\middle|p(y|0)\hat{p}_X(0) + \sum_{x\in\mathcal{X}^+} p(y|x)\hat{p}_X(x) + \sum_{x\in\mathcal{X}^-} p(y|x)\hat{p}_X(x)\right] \\
&= I\left[p_Y(y)\middle|\middle|p(y|0)\hat{p}_X(0) + \sum_{x\in\mathcal{X}^+} \left\{p(y|x)\hat{p}_X(x) + p(y|-x)\hat{p}_X(-x)\right\}\right] \\
&= I\left[p_Y(y)\middle|\middle|p(y|0)\hat{p}_X(0) + \sum_{x\in\mathcal{X}^+} \left\{p(y|x)\hat{p}_X(x) + p(y|-x)\hat{p}_X(x)\right\}\right] \\
&= I\left[p_Y(y)\middle|\middle|p(y|0)\hat{p}_X(0) + \sum_{x\in\mathcal{X}^+} \left\{p(y|x) + p(y|-x)\right\}\hat{p}_X(x)\right],
\end{aligned} \tag{177}$$

where $\mathcal{X}^+$ and $\mathcal{X}^-$ are defined as

$$\begin{aligned}
\mathcal{X}^+ &= \{x \in \mathcal{X} : x > 0\}, \\
\mathcal{X}^- &= \{x \in \mathcal{X} : x < 0\}.
\end{aligned} \tag{178}$$

The second to last equality holds by the assumption of the symmetry of the input density. Inspired by the structure of (177), we define a new kernel[2]

$$q(y|x) = \begin{cases} p(y|x) + p(y|-x), & x > 0 \\ p(y|0), & x = 0 \\ 0, & x < 0 \end{cases}. \tag{179}$$

Note that this kernel is still nonnegative, since $q(y|x)$ is defined by adding two elements in $p(y|x)$, which are nonnegative. Hence, we can still apply the (original) minimum I-divergence algorithm. With this new kernel, however, we only need to work on the estimate

[2]Here we deal with the case where the number of input samples is odd. Nevertheless, other researchers should be able to apply our method easily to the case of an even number of samples by following the procedure described in this section.

values for $x \in \mathcal{X}^+ \bigcup\{0\}$. Following the same procedure by which the original algorithm in Section 8.2.1 is derived, we find a new algorithm:

$$\hat{p}_X^{(k+1)}(x) = \frac{\hat{p}_X^{(k)}(x)}{Q(x)} \sum_{y \in \mathcal{Y}} \left[\frac{q(y|x) p_Y(y)}{\sum_{x' \in \mathcal{X}^+ \bigcup\{0\}} q(y|x') \hat{p}_X^{(k)}(x')} \right], \qquad \forall x \in \mathcal{X}^+ \bigcup\{0\}, \tag{180}$$

where

$$\begin{aligned} Q(x) &= \sum_{y \in \mathcal{Y}} q(y|x) \\ &= \begin{cases} \sum_{y \in \mathcal{Y}} p(y|x) + p(y|-x), & x > 0 \\ \sum_{y \in \mathcal{Y}} p(y|0), & x = 0 \end{cases} \\ &= \begin{cases} 2, & x > 0 \\ 1, & x = 0 \end{cases}. \end{aligned} \tag{181}$$

Specifically, this algorithm minimizes Csiszár's I-divergence between the known output $p_Y(y)$ and the output $\tilde{p}_Y(y)$ produced by a new system defined as

$$\tilde{p}_Y(y) = \sum_{x \in \mathcal{X}^+ \bigcup\{0\}} q(y|x) \hat{p}(x), \tag{182}$$

where the estimate for $x \in \mathcal{X}^-$ is tentatively assumed to be zero. Once the whole estimate for $x \in \mathcal{X}^+ \bigcup\{0\}$ is obtained, the estimate for $x \in \mathcal{X}^-$ is defined as

$$\hat{p}_X^{(k+1)}(x) = \hat{p}_X^{(k+1)}(-x) \qquad \forall x \in \mathcal{X}^-. \tag{183}$$

All the properties and relative theorems [107] still hold for the proposed algorithm associated with the newly defined system.

8.2.2.2 Symmetry of the Output

The symmetry of the output allows an additional improvement of computational efficiency. Let

$$\tilde{p}_Y^{(k)}(y) = \sum_{x' \in \mathcal{X}^+ \bigcup\{0\}} q(y|x') \hat{p}_X^{(k)}(x'), \tag{184}$$

and

$$r_Y^{(k)}(y) = \frac{p_Y(y)}{\hat{p}_Y^{(k)}(y)}. \tag{185}$$

Then, the algorithm (180) becomes

$$\hat{p}_X^{(k+1)}(x) = \frac{\hat{p}_X^{(k)}(x)}{Q(x)} \sum_{y \in \mathcal{Y}} q(y|x) r_Y^{(k)}(y),$$

$$\forall x \in \mathcal{X}^+ \bigcup \{0\}. \tag{186}$$

Recall that since the input $p_X(x)$ is symmetric, so are $p_Y(y)$ and $\hat{p}_Y(y)$, and in turn so is $r_Y(y)$, namely $r_Y(y) = r_Y(-y)\ \forall y \in \mathcal{Y}$. Using this symmetry, (186) can be rewritten as

$$\hat{p}_X^{(k+1)}(x) = \frac{\hat{p}_X^{(k)}(x)}{Q(x)} \left[\sum_{y \in \mathcal{Y}^+} \{q(y|x) + q(-y|x)\} r^{(k)}(y) + q(0|x) r^{(k)}(0) \right],$$

$$\forall x \in \mathcal{X}^+ \bigcup \{0\}, \tag{187}$$

where $\mathcal{Y}^+$ is defined as $\mathcal{Y}^+ = \{y \in \mathcal{Y} : y > 0\}$. Let

$$s(y|x) = \begin{cases} q(y|x) + q(-y|x), & y > 0 \\ q(0|x), & y = 0 \\ 0, & y < 0 \end{cases}. \tag{188}$$

Also, let $|\mathcal{Y}^+|$ denote the cardinality of $\mathcal{Y}^+$. Then (187) becomes

$$\hat{p}_X^{(k+1)}(x) = \frac{\hat{p}_X^{(k)}(x)}{S(x)} \sum_{y \in \mathcal{Y}^+ \bigcup \{0\}} s(y|x) r^{(k)}(y),$$

$$\forall x \in \mathcal{X}^+ \bigcup \{0\}, \tag{189}$$

where

$$
\begin{aligned}
S(x) &= \sum_{y \in \mathcal{Y}^+ \bigcup \{0\}} s(y|x) \\
&= \begin{cases} \sum_{y \in \mathcal{Y}^+} s(y|x) + s(0|x), & x > 0 \\ \sum_{y \in \mathcal{Y}^+} s(y|0) + s(0|0), & x = 0 \end{cases} \\
&= \begin{cases} \sum_{y \in \mathcal{Y}^+} \{q(y|x) + q(-y|x)\} + q(0|x), & x > 0 \\ \sum_{y \in \mathcal{Y}^+} \{q(y|0) + q(-y|0)\} + q(0|0), & x = 0 \end{cases} \\
&= \begin{cases} \sum_{y \in \mathcal{Y}^+} \{p(y|x) + p(y|-x) + p(-y|x) + p(-y|-x)\} + p(0|x) + p(0|-x), & x > 0 \\ \sum_{y \in \mathcal{Y}^+} \{p(y|0) + p(-y|0)\} + p(0|0), & x = 0 \end{cases} \\
&= \begin{cases} 2 - p(0|x) - p(0|-x) + p(0|x) + p(0|-x), & x > 0 \\ 1 - p(0|0) + p(0|0), & x = 0 \end{cases} \\
&= \begin{cases} 2, & x > 0 \\ 1, & x = 0 \end{cases} .
\end{aligned} \tag{190}
$$

In (189), the number of multiplications in the summation in the numerator decreases to about half the number in (187). Again, the remaining part of the estimate can be obtained by (183).

This new algorithm can be implemented as follows:

1. Begin with a feasible nonnegative, symmetric input estimate $\hat{p}_X^{(0)}(x)$, $\forall x \in \mathcal{X}^+ \bigcup \{0\}$.

2. Divide the known output density by the estimated output $\tilde{p}_Y^{(k)}(y)$ to obtain $r_Y^{(k)}(y)$, $\forall y \in \mathcal{Y}^+ \bigcup \{0\}$.

3. Compute the term in (189) where the summation over y is calculated. Call this $V^{(k)}(x) = \sum_{y \in \mathcal{Y} \bigcup \{0\}} s(y|x) r_Y^{(k)}(y)$.

4. Update the estimate of $\hat{p}_X(x)$ by

$$\hat{p}_X^{(k+1)}(x) = \frac{1}{S(x)} \hat{p}_X^{(k)}(x) V^{(k)}(x). \tag{191}$$

5. Repeat steps 2 through 4 until a convergence criterion is met.

6. Complete the estimate using (183).

8.2.3 Equivalence of the Algorithms

8.2.3.1 Symmetry of the Estimates

This section shows that if the transition kernel is centrosymmetric (as in the Rician case), then the the original minimum *I*-divergence algorithm and our symmetry-preserving minimum *I*-divergence algorithm are essentially the same, provided that both the algorithms are initialized with the same symmetric density. In other words, if the initial estimate is symmetric and the kernel has the desired symmetry, the original minimum *I*-divergence algorithm preserves the symmetry of the input density estimates, and the estimates from the two algorithms at each iteration are the same.

We first show that when $\hat{p}_X^{(k)}(x)$ is symmetric and the kernel has the desired properties, then the original minimum *I*-divergence algorithm induces the same symmetry on $\hat{p}_X^{(k+1)}(x)$. If the algorithm is initialized with a symmetric density $\hat{p}_X^{(0)}(x)$ (such as a uniform density), then mathematical induction implies that the original minimum *I*-divergence algorithm preserves the symmetry for all k. First, suppose that $\hat{p}_X^{(k)}(x)$ is symmetric, and suppose the kernel is $p(y|x)$ is symmetric with respect to the origin. This is clearly the case for the Rician kernel:

$$\begin{aligned} p(-y|-x) &= \frac{1}{\sqrt{2\pi(\sigma_h^2 x^2 + \sigma_n^2)}} e^{-\frac{1}{2(\sigma_h^2 x^2+\sigma_n^2)}(-y+\bar{h}x)^2} \\ &= \frac{1}{\sqrt{2\pi(\sigma_h^2 x^2 + \sigma_n^2)}} e^{-\frac{1}{2(\sigma_h^2 x^2+\sigma_n^2)}(y-\bar{h}x)^2} = p(y|x). \end{aligned} \tag{192}$$

Then, the estimate of the output $\hat{p}_Y^{(k)}(y)$ is symmetric because

$$\begin{aligned} \hat{p}_Y^{(k)}(-y) &= \sum_{x\in\mathcal{X}} p(-y|x)\hat{p}_X^{(k)}(x) = \sum_{x\in\mathcal{X}} p(y|-x)\hat{p}_X^{(k)}(x) \\ &= \sum_{x'\in\mathcal{X}} p(y|x')\hat{p}_X^{(k)}(-x') = \sum_{x'\in\mathcal{X}} p(y|x')\hat{p}_X^{(k)}(x') = \hat{p}_Y^{(k)}(y), \end{aligned} \tag{193}$$

where $x' = -x$. The second equality holds by the symmetry of the kernel in (192), and the fourth equality holds by the symmetry of the estimate of the input $\hat{p}_X^{(k)}(x)$. Furthermore, in (176), $W^{(k)}$ is symmetric since

$$\begin{aligned} W^{(k)}(-x) &= \sum_{y\in\mathcal{Y}} p(y|-x)\frac{p_Y(y)}{\hat{p}_Y^{(k)}(y)} = \sum_{y\in\mathcal{Y}} p(-y|x)\frac{p_Y(y)}{\hat{p}_Y^{(k)}(y)} \\ &= \sum_{y'\in\mathcal{Y}} p(y'|x)\frac{p_Y(-y')}{\hat{p}_Y^{(k)}(-y')} = \sum_{y'\in\mathcal{Y}} p(y'|x)\frac{p_Y(y')}{\hat{p}_Y^{(k)}(y')} = W^{(k)}(x), \end{aligned} \tag{194}$$

where $y' = -y$. The second equality holds by the symmetry of the kernel in (192), and the fourth equality holds by the symmetry of the true output and the output derived from $\hat{p}_X(x)$, as proven in (193). Consequently, by the assumption that $\hat{p}_X^{(k)}(x)$ is symmetric, the $(k+1)^{st}$ estimate of the input given by (176) is symmetric:

$$\hat{p}_X^{(k+1)}(-x) = \hat{p}_X^{(k)}(-x)W^{(k)}(-x) = \hat{p}_X^{(k)}(x)W^{(k)}(x) = \hat{p}_X^{(k+1)}(x). \tag{195}$$

8.2.3.2 Equivalence of the Iterations

Next, we show that the estimates from the two algorithms at each iteration are the same. Note that the algorithms in (186) and (189) are the same since the algorithm in (186) has been modified while maintaining the mathematical equivalence of each step. Hence, it is sufficient to show the equivalence of the algorithms in (175) and (186). Since the kernel $p(y|x)$ is centrosymmetric (as seen in Fig. 100), the new kernel $q(y|x)$ possesses one-dimensional symmetry with respect to the x-axis:

$$q(y|x) = p(y|x) + p(y|-x) = p(-y|-x) + p(-y|x) = q(-y|x). \tag{196}$$

Then, the algorithm in (186) can be rewritten as follows:

$$\begin{aligned} \hat{p}^{(k+1)}(x) &= \begin{cases} \dfrac{\hat{p}^{(k)}(x)}{2}\displaystyle\sum_{y\in\mathcal{Y}} q(y|x)r^{(k)}(y), & x>0 \\ \hat{p}^{(k)}(0)\displaystyle\sum_{y\in\mathcal{Y}} q(y|0)r^{(k)}(y), & x=0 \end{cases} \\ &= \begin{cases} \dfrac{\hat{p}^{(k)}(x)}{2}\left[\displaystyle\sum_{y\in\mathcal{Y}^+} q(y|x)r^{(k)}(y) + \sum_{y\in\mathcal{Y}^-} q(y|x)r^{(k)}(y) + q(0|x)r^{(k)}(0)\right], & x>0 \\ \hat{p}^{(k)}(0)\displaystyle\sum_{y\in\mathcal{Y}} p(y|0)r^{(k)}(y), & x=0 \end{cases}. \end{aligned} \tag{197}$$

In (197), when $x = 0$, it is clear that if the algorithms (175) and (186) are initialized with the same symmetric density, they produce the same estimates at each iteration, because

$$\begin{aligned}
\tilde{p}_Y^{(k)}(y) &= \sum_{x' \in \mathcal{X}^+ \bigcup \{0\}} q(y|x')\hat{p}_X^{(k)}(x') \\
&= \sum_{x' \in \mathcal{X}^+} \{p(y|x') + p(y|-x')\}\hat{p}_X^{(k)}(x') + p(y|0)\hat{p}_X^{(k)}(0) \\
&= \sum_{x' \in \mathcal{X}^+} p(y|x')\hat{p}^{(k)}(x') + \sum_{x' \in \mathcal{X}^+} p(y|-x')\hat{p}^{(k)}(-x') + p(y|0)\hat{p}_X^{(k)}(0) \\
&= \sum_{x' \in \mathcal{X}^+} p(y|x')\hat{p}^{(k)}(x') + \sum_{x'' \in \mathcal{X}^-} p(y|x'')\hat{p}^{(k)}(x'') + p(y|0)\hat{p}_X^{(k)}(0) \\
&= \sum_{x \in \mathcal{X}} p(y|x)\hat{p}_X^{(k)}(x) = \hat{p}_Y^{(k)}(y), \qquad (198)
\end{aligned}$$

where $x'' = -x'$, and hence it follows that

$$r_Y^{(k)}(y) = \frac{p_Y(y)}{\tilde{p}_Y^{(k)}(y)} = \frac{p_Y(y)}{\hat{p}_Y^{(k)}(y)}. \qquad (199)$$

The second equality holds by the definition of $q(y|x)$, and the third equality holds by the symmetry of $\hat{p}^{(k)}(x)$. When x is positive, the algorithm in (197) can be rewritten as follows:

$$\begin{aligned}
\hat{p}^{(k+1)}(x) &= \frac{\hat{p}^{(k)}(x)}{2}\left[\sum_{y \in \mathcal{Y}^+} q(y|x)r^{(k)}(y) + \sum_{y \in \mathcal{Y}^-} q(y|x)r^{(k)}(y) + q(0|x)r^{(k)}(0)\right] \\
&= \frac{\hat{p}^{(k)}(x)}{2}\left[\sum_{y \in \mathcal{Y}^+} q(y|x)r^{(k)}(y) + \sum_{y' \in \mathcal{Y}^+} q(y'|x)r^{(k)}(y') + q(0|x)r^{(k)}(0)\right] \\
&= \frac{\hat{p}^{(k)}(x)}{2}\left[2\sum_{y \in \mathcal{Y}^+} \{p(y|x) + p(y|-x)\}r^{(k)}(y) + \{p(0|x) + p(0|-x)\}r^{(k)}(0)\right] \\
&= \hat{p}^{(k)}(x)\left[\sum_{y \in \mathcal{Y}^+} \{p(y|x) + p(-y|x)\}r^{(k)}(y) + p(0|x)r^{(k)}(0)\right] \\
&= \hat{p}^{(k)}(x)\left[\sum_{y \in \mathcal{Y}^+} p(y|x)r^{(k)}(y) + \sum_{y' \in \mathcal{Y}^-} p(y'|x)r^{(k)}(y') + p(0|x)r^{(k)}(0)\right] \\
&= \hat{p}^{(k)}(x)\sum_{y \in \mathcal{Y}} p(y|x)r^{(k)}(y), \qquad (200)
\end{aligned}$$

where $y' = -y$. The second equality holds by the symmetries of $q(y|x)$ and $r^{(k)}(y)$, the fourth equality holds because $p(0|x) = p(0|-x)$, and the second to last equality holds by the symmetry of $r^{(k)}(y)$. Note that, using the relation in (199), the last term in (200) is the exactly same as the algorithm in (175). Combining this finding with the preservation of the symmetry of the estimates from the original minimum I-divergence algorithm, we conclude that the original algorithm and the modified algorithms produce the same estimate at each iteration if they are initialized with the same symmetric function.

It should be noted that the equivalence of the original algorithm to the modified algorithms holds only under certain conditions and the special structure of the current application such as the symmetry of the kernel. Our new algorithms (186) and (189), however, can preserve the symmetry of the input even if the conditions for the equivalence (i.e. such as the symmetry of the kernel) do not hold. The estimates from the original algorithm and the proposed symmetry-preserving algorithms are not necessarily the same in general.

8.3 Simulations

This section presents numerical results. In our simulation study, we adhere to the estimation of symmetric input densities because the authors of [37] have concluded that the input density for a Rician channel should be symmetric. To help develop a feel for the nature of the Rician kernel, we plot it for various sets of parameters. Among the parameter sets, two interesting cases are selected to compare the behaviors of the sequence of the estimates. Since we have proven that the original minimum I-divergence algorithm and the symmetry-preserving minimum I-divergence algorithm for these kernels are equivalent on the initial conditions, we only show the results from the symmetry-preserving minimum I-divergence algorithm. For the comparison of the computational efficiency, statistics of the execution times of the two algorithms are given.

In the first set of studies presented in this section, a known input distribution is reconstructed to test the algorithm. Of course, in practice, a researcher will posit some output distribution and seek an associated input distribution. It would be interesting to see what input distribution estimate our algorithm reports *when there is no input distribution that exactly gives the posited output distribution.* An interesting rectangular output is suggested, and an input density that generates a *closest* output is estimated and discussed.

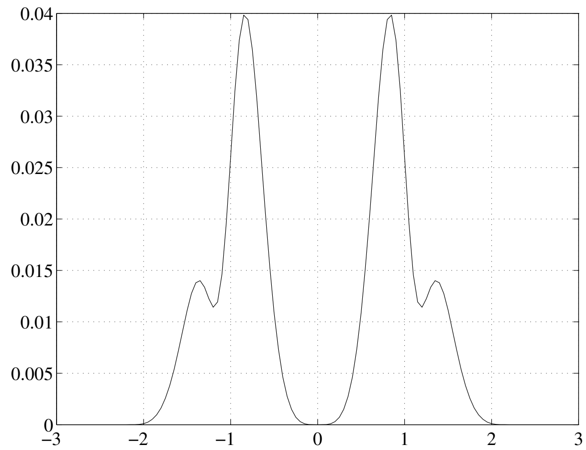


Figure 97: Symmetric channel input density .

Iterative reconstruction algorithms often suffer from various artifacts. As discussed in [104], these include *noise artifacts* and *edge artifacts,* which have been observed in emission tomography [105]. We are assuming that the "data" for the algorithm, namely the output distribution, are being proposed by a researcher, and hence noise is not a problem in the usual sense. "Edge artifacts" may arise when the kernel attenuates high frequency contents to the point that they cannot be reconstructed due to finite machine precision. Our Rician kernel certainly has this property; hence, we report how *edge artifacts* may be manifested by our algorithms.

8.3.1 Investigation of the Kernel

Figure 97 shows a symmetric input density. In this symmetric density, the $x \geq 0$ part is made by summing two differently weighted and shifted Weibull pdfs whose *characteristic life* and *shape* parameters are 2 and 5, respectively. Then, the $x < 0$ part is determined by symmetry. In implementation, the intermediate vector variables such as $W^{(k)}(x)$ in (176) and $V^{(k)}(x)$ in (191) are set to be longer than the input to avoid loss of information due to the support spreading of the data by the convolution-like operations in the algorithm. We must use the term "like," since the blurring is not technically a convolution since it is not space-invariant. In Fig. 97, only part of the x-axis is shown to best show details of the function.

Figure 98 shows a contour plot of the transition kernel parameterized with $\sigma_h = 0.5$,

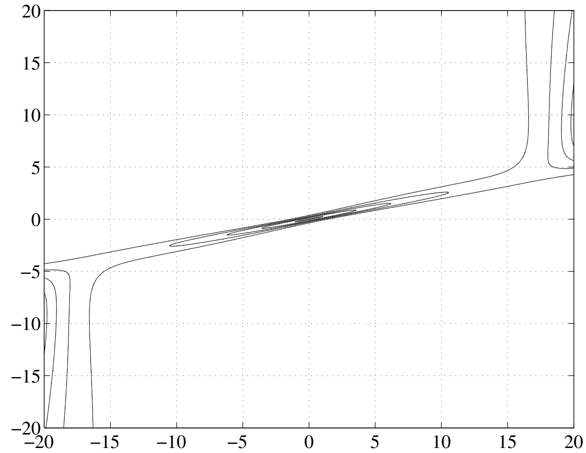


Figure 98: Contour plot of the kernel parameterized with $\sigma_h = 0.5$, $\sigma_n = 0.6$, and $\overline{h} = 4$. The horizontal axis is associated with y (the column of a transition matrix), and the vertical axis is associated with x (the row of a transition matrix)

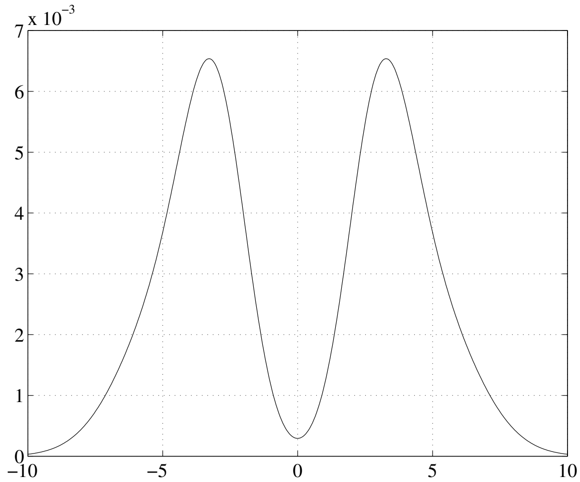


Figure 99: Output induced by the transition kernel in Fig. 98 given the input in Fig. 97.

$\sigma_n = 0.6$, and $\overline{h} = 4$. Because the shape of the transition area of the kernel can be grasped better with the contour plot than with a three-dimensional mesh plot, just the contour plot is displayed. Figure 99 shows an output induced by the transition kernel in Fig. 98 given the input in Fig. 97. Note that the output is symmetric as mentioned in [37]. The output is somewhat blurred; it loses some important features of the input. To best show details, only a portion of the x-axis is shown (as in Fig. 99).

A broader transition kernel induces a blurrier output density. A blurrier output may cause "slower" convergence of the estimates than a less blurry output. Therefore, the convergence behavior of the estimates is highly dependent on the shape of the kernel. The shape of the kernel is configured by the parameters, σ_h, σ_n, and $\overline{h}$. Hence, we show the effects of the parameters on the shape of the kernel by showing changes of the shape induced by changes of the parameters. In our description, we regard the kernel in Fig. 98 as a standard. The effects of each parameter on the shape of the kernel are illustrated by setting the value of the parameter larger or smaller than the value of the same parameter used in Fig. 98 while leaving the other two parameters fixed. Figures 100(a) and 100(b), 100(c) and 100(d), and 100(e) and 100(f) show the kernels resulting from larger and smaller values of σ_h, σ_n, and $\overline{h}$, respectively. For instance, Figs. 100(a) and 100(b) show kernels parameterized with $\sigma_h = 0.1$ and $\sigma_h = 0.9$, respectively, while the other parameters $\sigma_n = 0.6$ and $\overline{h} = 4$ remain the same for both.

Compare Fig. 98 with Fig. 100, and Fig. 99 with Fig. 101. It may be reasonable to conclude that the effects of σ_h and σ_n on the shape of the kernel, hence the outputs, are not significant. More specifically, the outputs in Figs. 101(a) and 101(b) do not show significant differences to the output in Fig. 99, although the kernels in 100(a), and 100(b) look somewhat different. Note that the transition areas near the center have similar sharpness. On the other hand, the kernels in Figs. 100(e) and 100(f) and their associated outputs shown in 101(c) are considerably different. The output associated with the broadest kernel (the kernel in Fig. 100(e)) is distinctively blurred, and the features of the input is completely destroyed. On the contrary, the output associated with the sharpest kernel (the kernel in Fig. 100(f)) is much less blurred and retain most of the features of the input, although it seems to lose the finest details, such as the two small side lobes (See Case II in Fig. 101(c)). Since the behavior of the estimates associated with the kernels in 100(a) through 100(d) are similar to the behavior of the estimates associated with the kernel in 100(f), we only show the estimates associated with the kernels used to get the outputs in 101(c).

8.3.1.1 Estimation Results for Known Channel Inputs

Figure 102 shows some estimates of the input density provided by the symmetry-preserving minimum *I*-divergence algorithm in (186). Recall that the estimates from the original

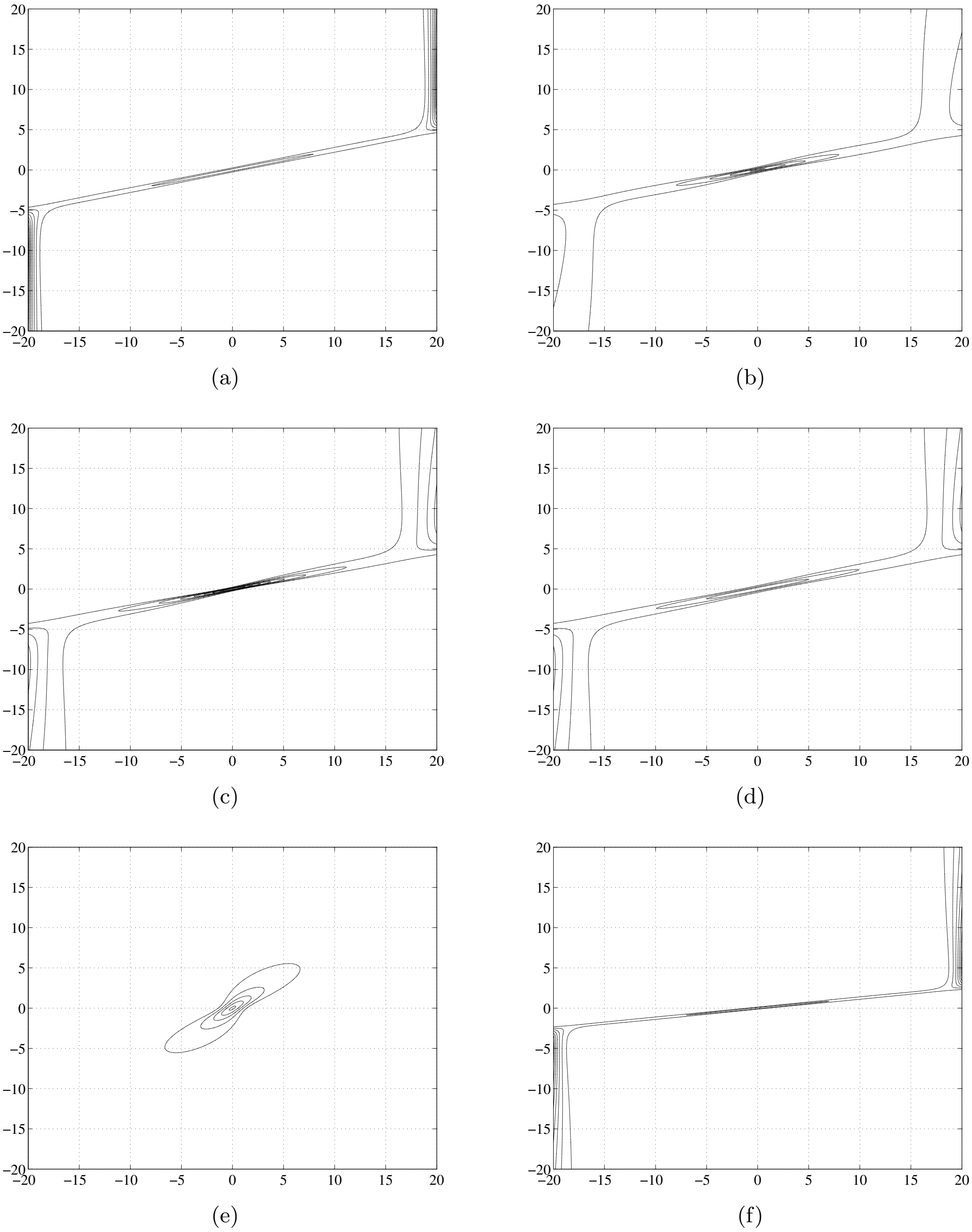


Figure 100: Contour plots of transition kernels for various choices of parameters: (a) $\sigma_h = 0.1$, $\sigma_n = 0.6$, and $\overline{h} = 4$. (b) $\sigma_h = 0.9$, $\sigma_n = 0.6$, and $\overline{h} = 4$. (c) $\sigma_h = 0.5$, $\sigma_n = 0.2$, and $\overline{h} = 4$. (d) $\sigma_h = 0.5$, $\sigma_n = 1.0$, and $\overline{h} = 4$. (e) $\sigma_h = 0.5$, $\sigma_n = 0.6$, and $\overline{h} = 1$. (f) $\sigma_h = 0.5$, $\sigma_n = 0.6$, and $\overline{h} = 8$.

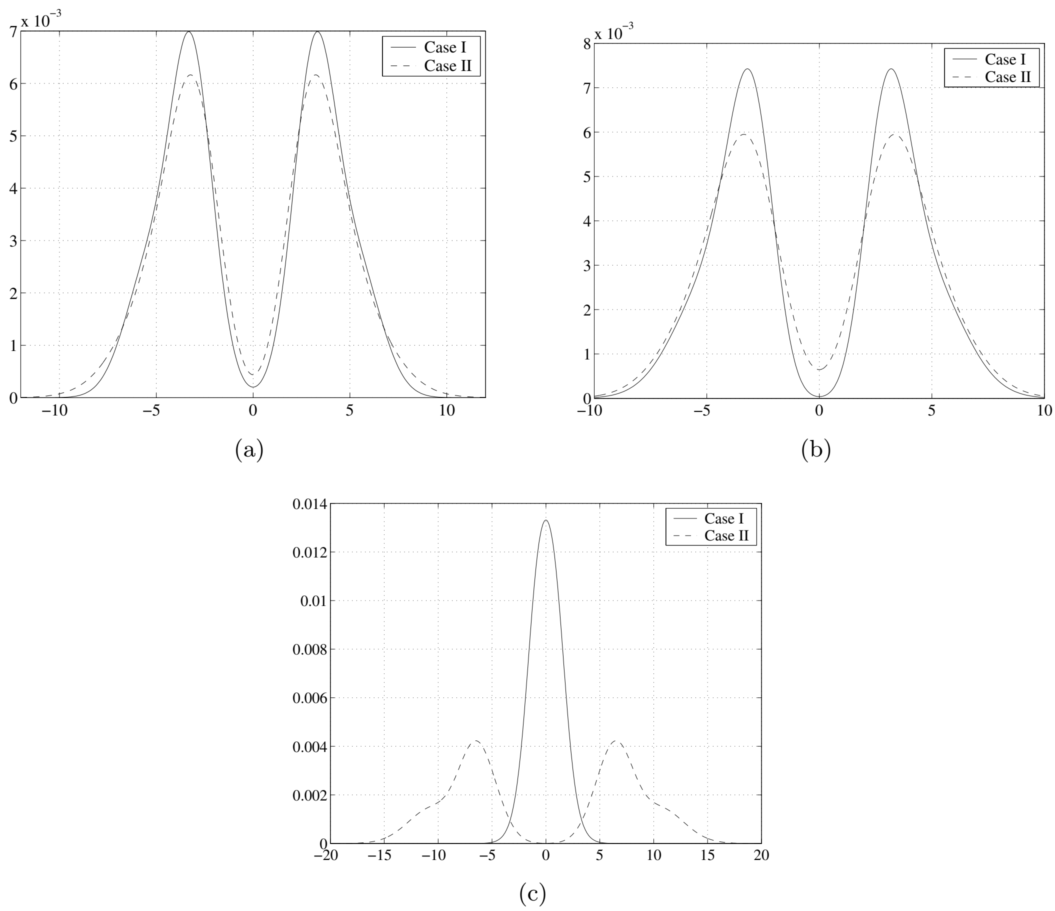


Figure 101: Channel outputs induced by the kernels given in Fig. 100 when the channel input in Fig. 97 is used: (a) Case I: $\sigma_h = 0.1$, $\sigma_n = 0.6$, and $\overline{h} = 4$; and Case II: $\sigma_h = 0.9$, $\sigma_n = 0.6$, and $\overline{h} = 4$. (b) Case I: $\sigma_h = 0.5$, $\sigma_n = 0.2$, and $\overline{h} = 4$; and Case II: $\sigma_h = 0.5$, $\sigma_n = 1.0$, and $\overline{h} = 4$. (c) Case I:$\sigma_h = 0.5$, $\sigma_n = 0.6$, and $\overline{h} = 1$; and Case II: $\sigma_h = 0.5$, $\sigma_n = 0.6$, and $\overline{h} = 8$.

minimum I-divergence algorithm and the symmetry-preserving minimum I-divergence algorithms are essentially the same, and hence we only show the estimates from the symmetry-preserving minimum I-divergence algorithm. The algorithms are initialized with the same uniform density. To show the behavior of the estimates, the figures of the estimates are shown for some selected iterations at which big transitions in the forms of the estimates appear. For example, the estimate at the 5^{th} iteration starts showing the two main lobes, and the estimate at the 20000^{th} starts showing the two small side lobes. Convergence of the estimates associated with the sharpest kernel and convergence of the estimates associated with the broadest kernel are in striking contrast with each other, as shown in Figs. 102(a) and 102(b), and 102(c) and 102(d). The estimates resulting from the least blurry output converge after 100 iterations, but the estimates from the blurriest output have not converged even after 20000 iterations. An important property of our algorithms is guaranteed convergence to the global optimum [107]. Based on this property and the observation that the algorithms take about 1800 iterations until the estimates show signs of the two small side lobes, it can be inferred that the estimates associated with the broadest kernel will converge after an enormous number of iterations.

All the experiments were performed with Matlab 6.5 by The Mathworks. The symmetry-preserving algorithm remarkably improves computation time. With the broadest kernel, the original minimum I-divergence algorithm takes 2808 seconds to reach the 20000^{th} iteration, but the symmetry-preserving minimum I-divergence algorithm given by (189) takes only 814 seconds.

8.3.2 Estimation Results for Arbitrary Specified Outputs

The previous section tested the algorithm by assuming a particular input, generating the outputs using the given kernel, and then watching the algorithm reconstruct the input. In practice, of course, a researcher will suggest an output, and an input that generates the exact output may not exist. In such cases, we aim at estimating an input that generates the *closest* output to the desired output in Csiszár's I-divergence sense. To demonstrate such an example, an ideal rectangle is proposed as the output distribution, and the resulting

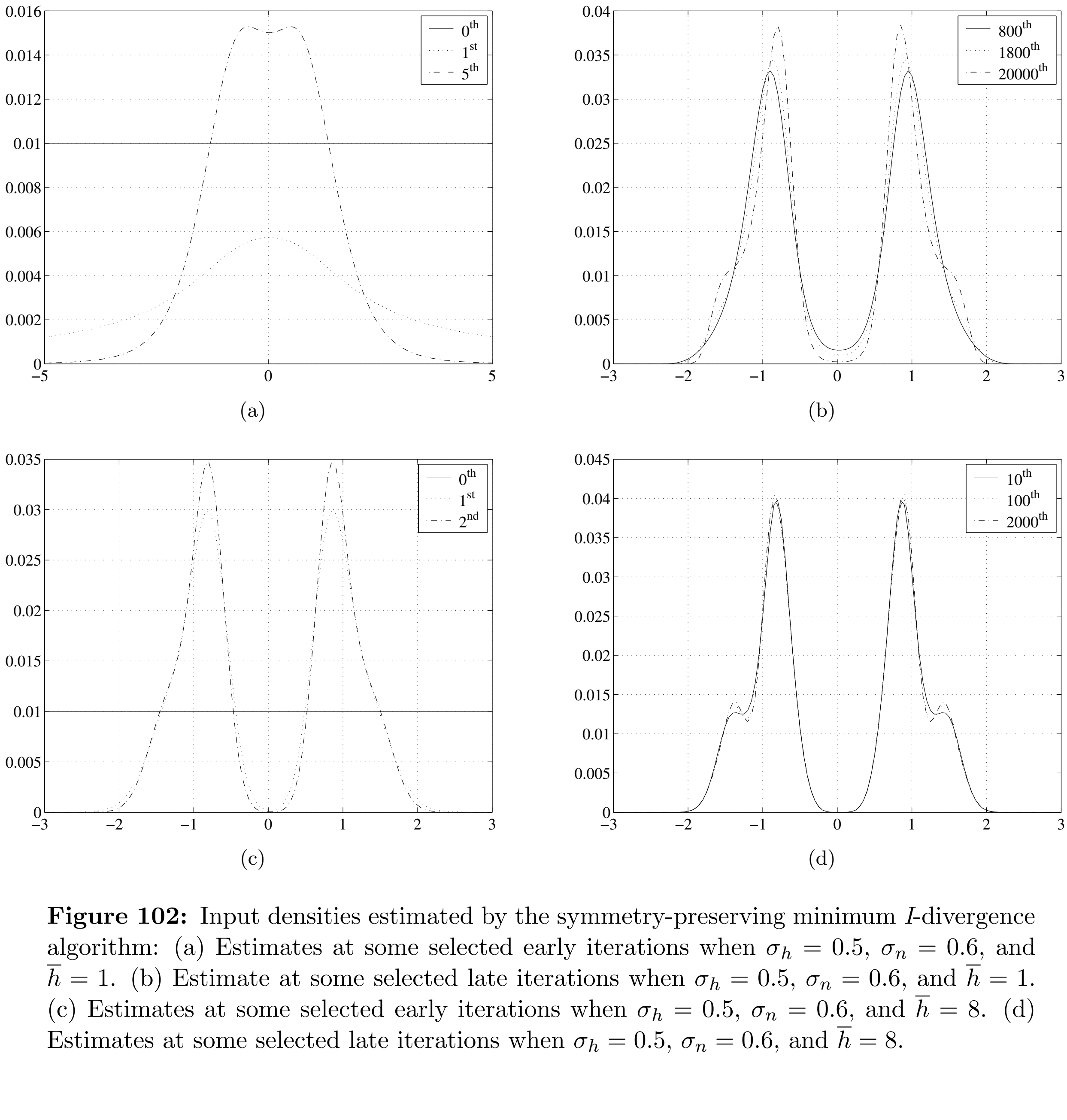


Figure 102: Input densities estimated by the symmetry-preserving minimum I-divergence algorithm: (a) Estimates at some selected early iterations when $\sigma_h = 0.5$, $\sigma_n = 0.6$, and $\overline{h} = 1$. (b) Estimate at some selected late iterations when $\sigma_h = 0.5$, $\sigma_n = 0.6$, and $\overline{h} = 1$. (c) Estimates at some selected early iterations when $\sigma_h = 0.5$, $\sigma_n = 0.6$, and $\overline{h} = 8$. (d) Estimates at some selected late iterations when $\sigma_h = 0.5$, $\sigma_n = 0.6$, and $\overline{h} = 8$.

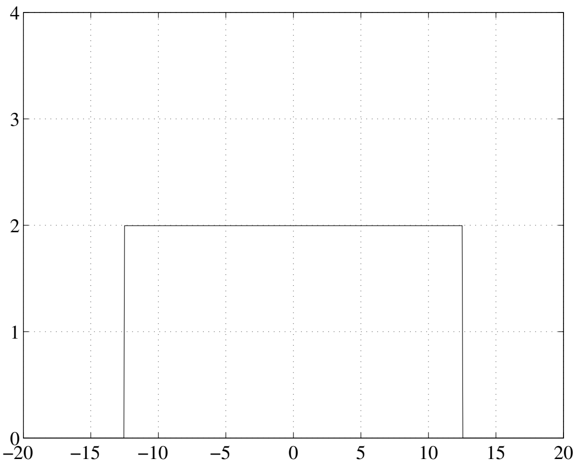


Figure 103: Ideal rectangle output density.

input estimates are shown and discussed.

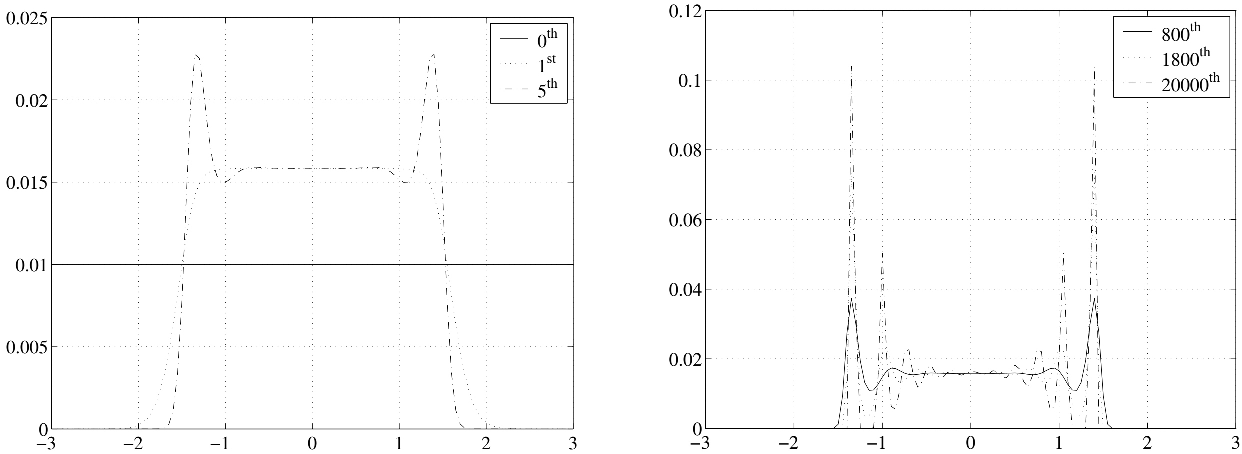
Figure 103 shows a desired rectangular output. Note that there is no channel input density that could induce this channel output because of the sharpness of the edges. Figure 104 shows the input estimate resulting from 20000 iterations of the algorithm. The input estimate is extremely spiky; interestingly, this is consistent with the guess made in [37]. Figure 105 shows the output density induced by the estimated input density. This estimate is the *closest* achievable output, in the sense that the *I*-divergence measure between the resulting output and the desired output is the smallest possible given the nonnegativity constraint on the input. As expected, the desired sharp edges cannot be obtained. Interestingly, the overshoot and ringing observed in the resulting *output* density are reminicent of the edge artifacts seen in the *input* estimates discussed in the following section.

8.3.3 The Edge Artifacts

8.3.3.1 Background on the Edge Artifacts

The problem of input distribution estimation may be thought of as a classic linear inverse problem. For discussion, let $\{\hat{p}(x) : x \in \mathcal{X}\}$ denote the function to be estimated, and $\{\mu(y) : y \in \mathcal{Y}\}$ denote the function observed or measured. Then the linear inverse problem can be represented as

$$\mu(y) = \int_{\mathcal{X}} p(y|x)\hat{p}(x)dx, \tag{201}$$



(a) (b)

Figure 104: Estimates of an input density generating the estimated output shown in Fig. 105. Early iterations are shown in (a), while later iterations are in (b).

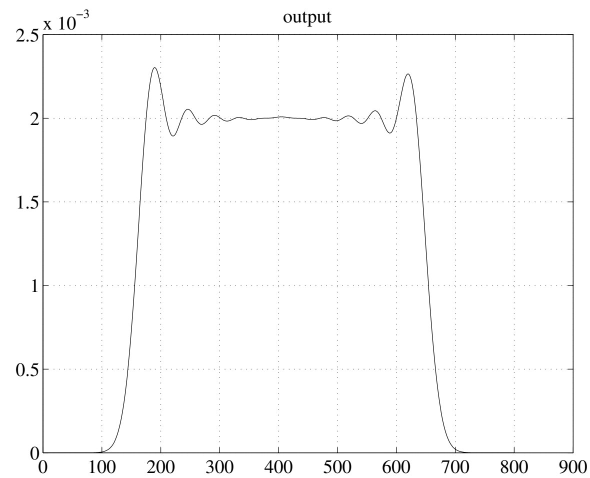


Figure 105: (a) Induced output closest to the output in Fig. 103 given the kernel in Fig. 100(f) is known.

where $p(y|x)$ is the transition kernel. These types of problems are notoriously unstable; small variations in $\mu(x)$ may cause relatively large variations in estimates of $\hat{p}(x)$. Such behavior may particularly affect the high frequency components in the estimates, and cause sharp transitions in the function being estimated to exhibit overshoot and ringing. These artifacts are called the *edge artifacts* [104] since high frequency components are usually distributed along edges. The *edge artifacts* are manifestations of Gibbs' phenomenon.

We will demonstrate how such artifacts appear in estimates by reconstructing a uniform function, which is often used for demonstrating these artifacts.

8.3.3.2 Demonstration of Edge Artifacts

Figure 106 shows an image of a uniform channel input density, and Figs. 107(a) and 107(b) show the outputs induced by the transition kernels in Figs. 100(e) and 100(f), respectively. Note that the outputs are symmetric. The *edge artifacts* are demonstrated in Fig. 108. The algorithm is initialized with a uniform density.

Figures 108(a) and 108(b), and 108(c) and 108(d) show the estimates of the uniform inputs given the outputs induced by the broadest kernel and the sharpest kernel, respectively. Recall that the estimates for the broadest kernel converge much slower than the estimates for the sharpest kernel. The estimates in Figs. 108(a) and 108(b) show the ringing artifacts first, and then the overshoots, which are gradually increasing. In contrast, the estimates in Fig. 108(c) and 108(d) show the relatively large overshoots first, and then start showing the ringings at later iterations.

8.4 Conclusion

We proposed a minimum I-divergence algorithm, along with new symmetry-preserving modifications, for estimating an input density given a kernel of a channel of interest and a target output density induced by the kernel. The rates of convergence depend on the shape of the kernel. The original algorithm and our proposed modifications are essentially equivalent at each iteration if the kernel and the initial estimate possess certain symmetries. The proposed symmetry-preserving algorithms provide considerable improvement in computation time.

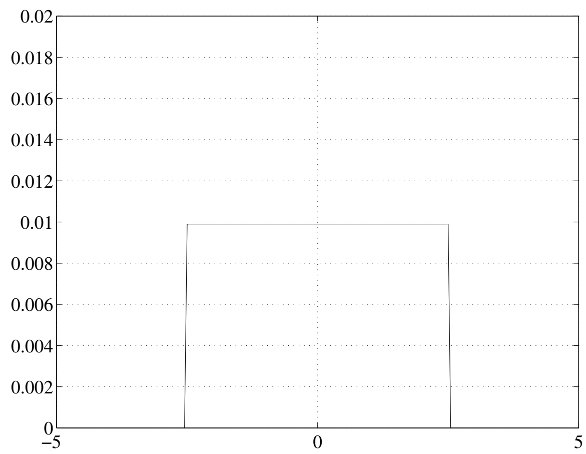


Figure 106: Symmetric uniform input density for demonstration of the *edge artifact*.

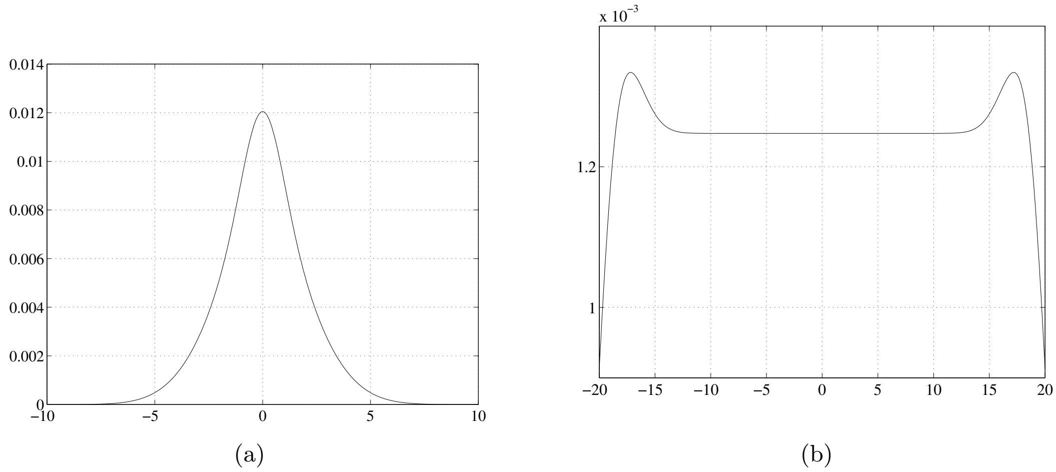


Figure 107: (a) Output corresponding to the input density in Fig. 106 when $\sigma_h = 0.5$, $\sigma_n = 0.6$, and $\overline{h} = 1$ are used. (b) Output for the input density in Fig. 106 when $\sigma_h = 0.5$, $\sigma_n = 0.6$, and $\overline{h} = 8$ are used.

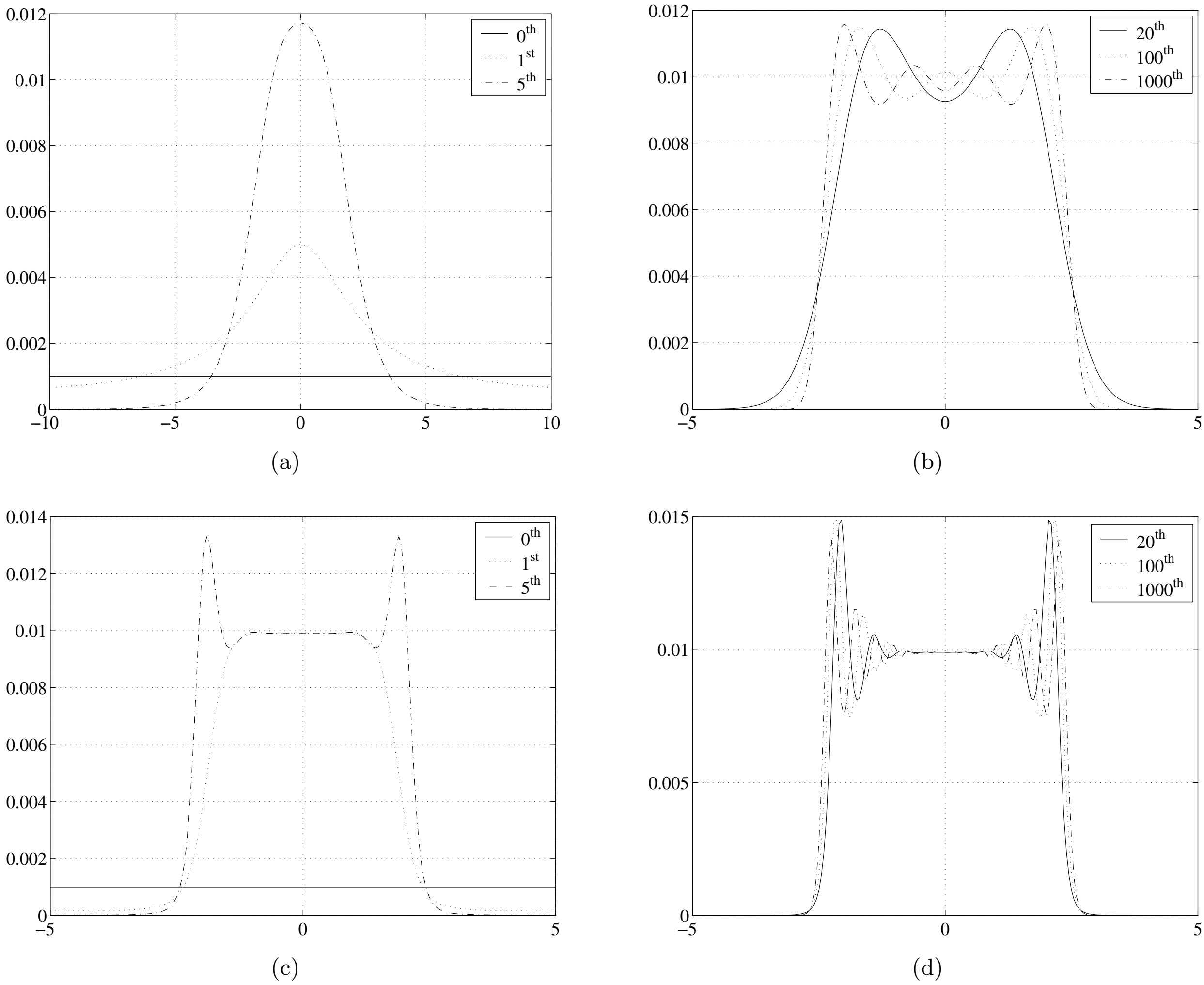


Figure 108: Estimates for the input density given in Fig. 106 reconstructed by the original minimum I-divergence algorithm: (a) Estimates at some selected early iterations when $\sigma_h = 0.5$, $\sigma_n = 0.6$, and $\overline{h} = 1$. (b) Estimates at some selected late iterations when $\sigma_h = 0.5$, $\sigma_n = 0.6$, and $\overline{h} = 1$. (c) Estimates at some selected early iterations when $\sigma_h = 0.5$, $\sigma_n = 0.6$, and $\overline{h} = 8$. (d) Estimates at some selected late iterations when $\sigma_h = 0.5$, $\sigma_n = 0.6$, and $\overline{h} = 8$.

Our simulation study has shown that the algorithms, both the original and its modifications, produce desirable results. The appearance of *edge artifacts* in our experiments was discussed, and regularization methods were breifly mentioned; their exploration remains an avenue for future work. An experiment of practical interest was performed in which an ideal output was proposed for which there exists no corresponding input. As expected, the algorithms found the input that results in an output as close as possible to the desired output.

The minimum *I*-divergence method has found use in a large number of applications. We have introduced another application of the method. We hope this research will stimulate interest in finding other applications of the minimum *I*-divergence approach in the communications literature.

CHAPTER IX

CONCLUSIONS

Numerous inverse problems in engineering and scientific applications have nonnegativity constraints. Csiszár's I-divergence defines an information-theoretic discrepancy measure between two nonnegative functions. When applied to inverse problems, I-divergences play the same role for nonnegative data as the squared error metrics play for real and complex data.

In the introduction, we reviewed an important conclusion by Csiszár [23]: minimizing the I-divergence is the only choice consistent with a set of intuitive postulates that are desirable in estimation. We also discuss how the I-divergence can be used for estimation subject to nonnegativity along with underlying structures to which the methods of minimizing I-divergence can be applied; the structures may be described by linear integral equations or convolution-like operations that may or may not be linear.

Our major application of minimum I-divergence methods was x-ray crystallography. Crystallography can be interpreted as a special case of phase retrieval. In Chapter 2, we proposed an iterative algorithm that tries to minimize I-divergence. This algorithm is a tweaked version of a phase retrieval algorithm invented by Schulz and Snyder [93] for their astronomical imaging application. The major difference between crystallography and astronomical imaging is that the autocorrelation is aliased in crystallography, which makes the problem much more difficult than in applications where the autocorrelations are not aliased. However, in crystallography, there are several potentially useful pieces of information such as space groups, which are special types of symmetries. We found that our tweaked algorithm can theoretically preserve a symmorphic group, which is a subcategory of the space groups, if the algorithm is initialized with an estimate with that symmorphic group. In practice, the space groups in the estimates produced by the algorithm are usually broken by numerical errors. Intriguingly, when the space groups are deliberately enforced at every iteration, such enforcement often leads the estimates to local minima.

In Chapter 3, we numerically studied an example of how the R-factor changes as the I-divergence monotonically decreases. Although we do not have any theoretical proofs on this matter, our numerical experiments with real data of $6PTI$ showed that an estimate sequence that monotonically decreased the I-divergence also decreased the R-factor.

The most challenging issue on the original Schulz-Snyder algorithm and our tweaked version is their potential convergence to local minima. In Chapter 4, we investigated the Schulz-Snyder algorithm's convergence to local minima via various pieces of numerical evidence. The algorithm turns out to sometimes converge to local minima even when measured autocorrelations are not aliased. We can easily infer that our tweaked version not only inherits this problem, but it also seems to be more vulnerable to this issue because of aliasing in the autocorrelations, as evidenced by some examples in Chapter 2.

Chapter 5 dealt with artifacts that arise when noise corrupts measurements. Like maximum-likelihood estimates in statistical contexts, our phase-retrieval estimates based on minimizing I-divergence are inclined to be rough. Noticeably, noise badly deteriorates our estimates suddenly when the noise level reaches a certain level, rather than gradually degrading as the signal-to-noise ratio becomes lower. By introducing Good's roughness and total variation penalties, we obtained reasonably smooth estimates. However, when the signal-to-noise ratio is significantly low, the penalties cannot help.

We noted that the deautoconvolution problem has a structure inspiringly similar to that of the deautocorrelation problem. We adapted the Schulz-Snyder idea for addressing deautocorrelation to deautoconvolution. The algorithms for both problems bear striking resemblances and similar theoretical properties. However, deautoconvolution is practically easier than deautocorrelation since there is much less ambiguity in the associated Fourier phases. Although numerical examples are promising, we could not provide proof of the iterative deautoconvolution algorithm's convergence to local minima (the same is true in Schulz and Snyder's original work [93]). This remains as an agenda for future work.

In Chapter 7, we addressed the inverse blackbody radiation problem using minimum I-divergence methods. The problem is ill-posed because of the characteristics of the kernel.

The ill-posedness becomes more serious when measurements are corrupted by noise. We formulated a penalized minimum I-divergence framework to handle this ill-posedness. When estimates are regularized, they show reasonable quality, but nevertheless, sufficient noise may render some information unrecoverable.

Moving to a communication problem, when an input distribution is blurred by a Rician channel to induce an output distribution, the channel input-output system can be characterized by the Fredholm equation of the first kind with a shift-varying kernel. Based on this, we derived an iterative algorithm for estimating a channel input from a desired corresponding channel output by minimizing I-divergence. Even though the convergence speed is slow, the estimates are good. When the suggested output is not actually achievable, we showed that our algorithm tries to find an estimate that produces an output closest to the suggested output in the sense of I-divergence.

Minimum (penalized-) I-divergence methods have been shown to be useful for both linear and nonlinear estimation problems subject to nonnegativity. Nonetheless, there remains some aspects to improve. In both linear and nonlinear problems, our iterative algorithms show slow convergence. We may be able to apply techniques for accelerating EM algorithms, such as the space-alternating EM algorithms proposed by Fessler and Hero [31]. When the I-divergence methods are applied to deautocorrelation, the iterative algorithms, in their current form, seriously suffer from the issue of convergence to local minima. This issue arises in other phase-retrieval problems, and is not unique to our methods. Global optimization techniques such as genetic algorithms or simulated annealing may be able to partially alleviate this challenge, although they lack the conceptual elegance of the EM-based algorithms.

APPENDIX A

SUPPLEMENTARY FOR CHAPTER II

A.1 Derivation of Algorithm 2

For the given problem, the estimate ρ_{est} that we want to find satisfies

$$\hat{\rho} = \arg\min_{\rho \geq 0} I(P||P_\rho), \tag{202}$$

where P and P_ρ are the measured Patterson function and the Patterson function of the estimate given in Eq. (19), and the notation $\rho \geq 0$ means that all components of ρ are nonnegative. The *I*-divergence $I(P||P_\rho)$ is given by

$$I(P||P_\rho) = \sum_{\mathbf{u}} \left\{ P(\mathbf{u}) \ln \frac{P(\mathbf{u})}{P_\rho(\mathbf{u})} + P_\rho(\mathbf{u}) - P(\mathbf{u}) \right\}. \tag{203}$$

Using the Kuhn-Tucker conditions [73], we obtain the necessary (but not sufficient) conditions for $\hat{\rho}$ to satisfy Eq. (202):

$$\frac{\partial I(P||P_\rho)}{\partial \hat{\rho}(\mathbf{r})} \begin{cases} = 0 & \hat{\rho}(\mathbf{r}) > 0 \\ \geq 0 & \hat{\rho}(\mathbf{r}) = 0 \end{cases}, \tag{204}$$

for all $\mathbf{r}$ in the unit cell. The first derivative of the *I*-divergence can be obtained as follows:

$$\begin{aligned} \frac{\partial I(P||P_\rho)}{\partial \hat{\rho}(\mathbf{r})} &= \sum_{\mathbf{u}} \left\{ P(\mathbf{u}) \frac{P_{\hat{\rho}}(\mathbf{u})}{P(\mathbf{u})} \frac{-P(\mathbf{u})}{[P_{\hat{\rho}}(\mathbf{u})]^2} \frac{\partial P_\rho(\mathbf{u})}{\partial \hat{\rho}} + \frac{\partial P_\rho(\mathbf{u})}{\partial \hat{\rho}} \right\} \\ &= \sum_{\mathbf{u}} \left\{ \frac{-P(\mathbf{u})}{P_{\hat{\rho}}(\mathbf{u})} \frac{\partial P_\rho(\mathbf{u})}{\partial \hat{\rho}} + \frac{\partial P_\rho(\mathbf{u})}{\partial \hat{\rho}} \right\}. \end{aligned} \tag{205}$$

Here, the first derivative of $P_\rho(\mathbf{u})$ can be expressed as follows:

$$\begin{aligned} \frac{\partial P_\rho(\mathbf{u})}{\partial \hat{\rho}} &= \frac{\partial}{\partial \hat{\rho}} \sum_{\mathbf{r}'} \rho((\mathbf{r}' + \mathbf{u}) \bmod \mathbf{d}) \rho(\mathbf{r}') \\ &= \hat{\rho}((\mathbf{r} + \mathbf{u}) \bmod \mathbf{d}) + \hat{\rho}(\mathbf{r} - \mathbf{u} + n\mathbf{d}) \\ &= \hat{\rho}((\mathbf{r} + \mathbf{u}) \bmod \mathbf{d}) + \hat{\rho}((\mathbf{r} - \mathbf{u}) \bmod \mathbf{d}), \end{aligned} \tag{206}$$

where the n in the second line is 0 if $\mathbf{r}+\mathbf{u}<\mathbf{d}$ and 1 otherwise. Plugging Eq. (206) into Eq. (205), we obtain

$$\begin{aligned}
\frac{\partial I(P||P_{\rho})}{\partial\hat{\rho}(\mathbf{r})} &= \sum_{\mathbf{u}}\left[\frac{-P(\mathbf{u})}{P_{\hat{\rho}}(\mathbf{u})}\left\{\hat{\rho}((\mathbf{r}+\mathbf{u}) \bmod \mathbf{d})+\hat{\rho}((\mathbf{r}-\mathbf{u}) \bmod \mathbf{d})\right\}\right] \\
&\quad +\sum_{\mathbf{u}}\left\{\hat{\rho}((\mathbf{r}+\mathbf{u}) \bmod \mathbf{d})+\hat{\rho}((\mathbf{r}-\mathbf{u}) \bmod \mathbf{d}\right\} \\
&= -2\sum_{\mathbf{u}}\hat{\rho}((\mathbf{r}+\mathbf{u}) \bmod \mathbf{d})\frac{P(\mathbf{u})}{P_{\hat{\rho}}(\mathbf{u})}+2\sum_{\mathbf{r}}\rho(\mathbf{r}) \\
&= -2\sum_{\mathbf{u}}\hat{\rho}((\mathbf{r}+\mathbf{u}) \bmod \mathbf{d})\frac{P(\mathbf{u})}{P_{\hat{\rho}}(\mathbf{u})}+2\left(\sum_{\mathbf{u}}P(\mathbf{u})\right)^{1/2}, \qquad (207)
\end{aligned}$$

where the second equality holds since Patterson functions are centrosymmetric. The last equality is satisfied by the following relation between $\sum_{\mathbf{u}} P(\mathbf{u})$ and $\sum_{\mathbf{r}}\hat{\rho}(\mathbf{r})$:

$$\begin{aligned}
\sum_{\mathbf{u}}P(\mathbf{u}) &= \sum_{\mathbf{u}}\sum_{\mathbf{r}}\hat{\rho}((\mathbf{r}+\mathbf{u}) \bmod \mathbf{d})\hat{\rho}(\mathbf{r}) \\
&= \sum_{\mathbf{r}}\hat{\rho}(\mathbf{r})\sum_{\mathbf{u}}\hat{\rho}((\mathbf{r}+\mathbf{u}) \bmod \mathbf{d}) \\
&= \sum_{\mathbf{r}}\hat{\rho}(\mathbf{r})\sum_{\mathbf{r}}\hat{\rho}(\mathbf{r}) \\
&= \left(\sum_{\mathbf{r}}\hat{\rho}(\mathbf{r})\right)^2 \qquad (208)
\end{aligned}$$

Setting Eq. (207) equal to zero suggests Algorithm 2:

$$\rho_{k+1}(\mathbf{r})=\rho_k(\mathbf{r})\frac{1}{\left(\sum_{\mathbf{u}}P(\mathbf{u})\right)^{1/2}}\sum_{\mathbf{u}}\rho_k((\mathbf{r}+\mathbf{u}) \bmod \mathbf{d})\frac{P(\mathbf{u})}{P_{\rho_k}(\mathbf{u})}. \qquad (209)$$

A.2 Proof of Theorem 1

We first justify Step *(S1)* in Theorem 1. Let $\mathcal{G}_t=\{(\mathbf{W}_k,\mathbf{0}):k=1,2,\ldots,J\}$ and $\mathcal{G}_i=\{(\mathbf{W}_k,\mathbf{w}_k):k=1,2,\ldots,J\}$, where the $\mathbf{W}_k$'s are common. $\mathcal{G}_i$ is the space group of ρ associated with the given P, and J is the number of elements of $\mathcal{G}_i$. As indicated by Eq.

(24), $|F|$ has the space group $\mathcal{G}_t$. Then, from Eq. (11), we obtain the following relations:

$$\begin{aligned} P(\mathbf{u}) &= \frac{1}{V}\sum_{\mathbf{h}} |F(\mathbf{h^T})|^2 \exp(-2\pi i \mathbf{h^T u}) \\ &= \frac{1}{V}\sum_{\mathbf{h}\in\mathcal{I}(\mathcal{G}_t)} \sum_{j=1}^{J} |F(\mathbf{h^T W}_j)|^2 \exp(-2\pi i \mathbf{h^T W}_j \mathbf{u}) \\ &= \frac{1}{V}\sum_{j=1}^{J} |F(\mathbf{h^T})|^2 \sum_{\mathbf{h}\in\mathcal{I}(\mathcal{G}_t)} \exp(-2\pi i \mathbf{h^T W}_j \mathbf{u}), \end{aligned} \tag{210}$$

where the second equality holds by Eq. (22), and the third equality holds by Eq. (24). Now, we want to show that

$$P(\mathbf{u}) = P(\mathbf{W}_l \mathbf{u}), \text{ for } l = 1, 2, \ldots, J, \tag{211}$$

where $(\mathbf{W}_l, 0) \in \mathcal{G}_t$. Let l be fixed. Plugging $\mathbf{h^T W}_l$ into Eq. (210), we observe that

$$P(\mathbf{W}_l \mathbf{u}) = \frac{1}{V}\sum_{j=1}^{J} |F(\mathbf{h^T})|^2 \sum_{\mathbf{h}\in\mathcal{I}(\mathcal{G}_t)} \exp(-2\pi i \mathbf{h^T W}_j \mathbf{W}_l \mathbf{u}). \tag{212}$$

Since $\mathbf{G}_j = (\mathbf{W}_j, \mathbf{0}) \in \mathcal{G}_t$ and $\mathbf{G}_l = (\mathbf{W}_l, \mathbf{0}) \in \mathcal{G}_t$, their composition also belongs to $\mathcal{G}_t$: $\mathbf{G}_j\mathbf{G}_l = (\mathbf{W}_j\mathbf{W}_l, \mathbf{0}) \in \mathcal{G}_t$, for all j. Furthermore, all such compositions will form $\mathcal{G}_t$, when l is fixed and each k is involved. Therefore, the right-hand side of Eq. (212) equals $P(\mathbf{u})$, and hence, Eq. (211) is satisfied. This space group $\mathcal{G}_t$ is a symmorphic group. However, a function with the space group $\mathcal{G}_t$ is not guaranteed to be centrosymmetric. Hence, at this stage, $\mathcal{G}_t$ is not necessarily complete for representing the symmetry of the Patterson function P.

Note that all translation components $\mathbf{w}_j$ have been removed. Equivalently, glide planes and/or screw axes have been replaced with the corresponding rotation axes and mirror planes.

Next, we justify Step ($\mathcal{S}2$). First, note that

$$P(-\mathbf{u}) = \sum_{\mathbf{r}} \rho((\mathbf{r}-\mathbf{u}) \mod \mathbf{d})\rho(\mathbf{r}). \tag{213}$$

Let $((\mathbf{r}-\mathbf{u}) \mod \mathbf{d}) = \mathbf{s}$, where $\mathbf{s}$ is a 3-D column vector. Then, $\mathbf{r} = \mathbf{s} + \mathbf{n^T d} + \mathbf{u}$, where $\mathbf{n}$ is a 3-D column vector whose elements are integers depending on $(\mathbf{r}-\mathbf{u})$. Rearranging this relation, we obtain $\mathbf{r} - \mathbf{n^T d} = \mathbf{s} + \mathbf{u}$. Taking the (mod $\mathbf{d}$) operation on both sides, we obtain

$\mathbf{r} = (\mathbf{r} - \mathbf{n}^{\mathbf{T}}\mathbf{d}) \mod \mathbf{d} = (\mathbf{s} + \mathbf{u}) \mod \mathbf{d}$. Consequently, we obtain the centrosymmetry of P:

$$
\begin{aligned}
P(-\mathbf{u}) &= \sum_{\mathbf{r}} \rho((\mathbf{r} - \mathbf{u}) \mod \mathbf{d})\rho(\mathbf{r}) \\
&= \sum_{\mathbf{s}} \rho(\mathbf{s})\rho((\mathbf{s} + \mathbf{u}) \mod \mathbf{d}) = P(\mathbf{u}).
\end{aligned} \tag{214}
$$

Using this centrosymmetry, Eq. (210) can be rewritten as

$$
\begin{aligned}
P(\mathbf{u}) &= P(-\mathbf{u}) \\
&= \frac{1}{V}\sum_{j=1}^{J} |F(\mathbf{h}^{\mathbf{T}})|^2 \sum_{\mathbf{h}\in\mathcal{I}(\mathcal{G}_t)} \exp(-2\pi i \mathbf{h}^{\mathbf{T}}\mathbf{W}_j(-\mathbf{u})) \\
&= \frac{1}{V}\sum_{j=1}^{J} |F(\mathbf{h}^{\mathbf{T}})|^2 \sum_{\mathbf{h}\in\mathcal{I}(\mathcal{G}_t)} \exp(-2\pi i \mathbf{h}^{\mathbf{T}}\mathbf{W}_j\mathbf{J}\mathbf{u}),
\end{aligned} \tag{215}
$$

where $\mathbf{J}$ is the 3×3 diagonal matrix whose elements are all -1. Thus, the group $\{(\mathbf{W}_j\mathbf{J}, \mathbf{0}) : k = 1, 2, \ldots, J\}$ can produce the same P. Hence, all the $(\mathbf{W}_j\mathbf{J}, \mathbf{0})$ should be added to the $\mathcal{G}_t$ obtained in Step ($\mathcal{S}1$), to form the Patterson space group (if these elements are not already in $\mathcal{G}_t$). These new group members $(\mathbf{W}_j\mathbf{J}, \mathbf{0})$ are simply the inversions (to avoid confusion, note these are *not* the matrix "inverses" of the matrices $\mathbf{W}_j$) of all the group members of $\mathcal{G}_t$ obtained in Step ($\mathcal{S}1$). Therefore, Step ($\mathcal{S}2$) is appropriately justified.

The final result is also a space group since it is also a "feasible" combination of the crystallographic symmetries, and the associated lattice system has not been destroyed. We call this resulting space group the Patterson space group and denote it by $\mathcal{H}_m \in \mathcal{PSG} \subset \mathcal{SSG} \subset \mathcal{SG}$.

A.3 Proof of Theorem 2

Recall that ρ has the space group $\mathcal{G}_i$, and $\mathbf{G}_j$, $j = 1, 2, \ldots, J$ denote the elements of $\mathcal{G}_i$. Let $\mathcal{G}_t$ denote the space group whose elements are $(\mathbf{W}_j, \mathbf{0})$, where $(\mathbf{W}_j, \mathbf{w}_j) \in \mathcal{G}_i$ for all $j = 1, 2, \ldots, J$.

We first show that if Algorithm 2 is initialized with a function with $\mathcal{G}_i$, and all the estimates produced by Algorithm 2 have the same space group $\mathcal{G}_i$, then all the translation components $\mathbf{w}_j$ for all $\mathbf{G}_j \in \mathcal{G}_i$ are zero: $\mathbf{w}_j = \mathbf{0}$ for all $j = 1, 2, \ldots, J$.

Assume that the algorithm is initialized with a function with $\mathcal{G}_i$. Also, assume that ρ_k for $k = 0, 1, \ldots$ have $\mathcal{G}_i$. Then, by Eq. (25), Q_k for $k = 0, 1, \ldots$ have $\mathcal{G}_i$ as well (*i.e.*, $Q_k(\mathbf{G}_j\mathbf{r}) = Q_k(\mathbf{r})$ for all $j = 1, 2, \ldots, J$).

Noting that Q_k consists of correlation operations, we look into Q_k through the Fourier relation as in our Patterson analysis in the proof of Theorem 1 in Section 2:

$$
\begin{aligned}
Q_k(\mathbf{r}) &= \frac{1}{V}\sum_{\mathbf{h}} \Lambda(\mathbf{h}^{\mathbf{T}}) \exp(-2\pi i \mathbf{h}^{\mathbf{T}}\mathbf{r}) \\
&= \frac{1}{V}\sum_{\mathbf{h}\in\mathcal{I}(\mathcal{G}_t)} \sum_{j=1}^{J} \Lambda(\mathbf{h}^{\mathbf{T}}\mathbf{W}_j) \exp(-2\pi i \mathbf{h}^{\mathbf{T}}\mathbf{W}_j\mathbf{r}) \\
&= \frac{1}{V}\sum_{\mathbf{h}\in\mathcal{I}(\mathcal{G}_t)} \sum_{j=1}^{J} \Lambda(\mathbf{h}^{\mathbf{T}}) \exp(2\pi i \mathbf{h}^{\mathbf{T}}\mathbf{w}_j) \exp(-2\pi i \mathbf{h}^{\mathbf{T}}\mathbf{W}_j\mathbf{r}), \\
&= \frac{1}{V}\sum_{\mathbf{h}\in\mathcal{I}(\mathcal{G}_t)} \Lambda(\mathbf{h}^{\mathbf{T}}) \sum_{j=1}^{J} \exp(-2\pi i (\mathbf{W}_j\mathbf{r} - \mathbf{w}_j)),
\end{aligned} \tag{216}
$$

where the second equality holds since $\mathbf{h}^{\mathbf{T}}\mathbf{W}_j$ is just a rearrangement of $\mathbf{h}^{\mathbf{T}}$, and the third equality holds by Eq. (27). In the second line, note that Λ does not have the space group $\mathcal{G}_t$ because of the exponential term adhering to $\Lambda(\mathbf{h}^{\mathbf{T}}\mathbf{W}_j)$ in Eq. (27). Nonetheless, we can classify $\mathbf{h}^{\mathbf{T}}$ and group $\Lambda(\mathbf{h}^{\mathbf{T}}\mathbf{W}_j)$ according to $\mathcal{G}_t$ since $\mathbf{h}^{\mathbf{T}}$ is only transformed by $\mathbf{W}_j$, and $\mathbf{w}_j$ in the exponential term has no effects on indices $\mathbf{h}^{\mathbf{T}}$. This justifies our use of $\mathcal{I}(\mathcal{G}_t)$ for Λ in the development of Eq. (216).

Recall that Q_k has space group $\mathcal{G}_i$ by the assumption that ρ_k has space group $\mathcal{G}_i$:

$$
\begin{aligned}
&Q_k(\mathbf{G}_l\mathbf{r}) = Q_k(\mathbf{r}), \text{ or} \\
&Q_k(\mathbf{W}_l\mathbf{r} + \mathbf{w}_l) = Q_k(\mathbf{r}), \text{ for all } \mathbf{G}_l \in \mathcal{G}_i.
\end{aligned} \tag{217}
$$

Replacing $\mathbf{r}$ with $\mathbf{W}_l\mathbf{r} + \mathbf{w}_l$ in Eq. (216), we obtain

$$
Q_k(\mathbf{W}_l\mathbf{r} + \mathbf{w}_l) = \frac{1}{V}\sum_{\mathbf{h}\in\mathcal{I}(\mathcal{G}_t)} \Lambda(\mathbf{h}^{\mathbf{T}}) \sum_{j=1}^{J} \exp(-2\pi i (\mathbf{W}_j\mathbf{W}_l\mathbf{r} + \mathbf{W}_j\mathbf{w}_l - \mathbf{w}_j)). \tag{218}
$$

For Eq. (217) to be satisfied, we should have

$$
\sum_{j=1}^{J} \exp(-2\pi i (\mathbf{W}_j\mathbf{W}_l\mathbf{r} + \mathbf{W}_j\mathbf{w}_l - \mathbf{w}_j)) = \sum_{j=1}^{J} \exp(-2\pi i (\mathbf{W}_j\mathbf{r} - \mathbf{w}_j)). \tag{219}
$$

Therefore, we need to find some condition(s) such that

$$\{(\mathbf{W}_j\mathbf{W}_l, \mathbf{W}_j\mathbf{w}_l - \mathbf{w}_j)\}_{j=1}^{J} = \{(\mathbf{W}_o, -\mathbf{w}_o)\}_{o=1}^{J}, \tag{220}$$

where l is fixed as an integer from 1 to J, $\mathbf{G}_l \in \mathcal{G}_i$, $\mathbf{G}_o = (\mathbf{W}_o, \mathbf{w}_o) \in \mathcal{G}_i$, and $\{\}_{j=1}^{J}$ denotes a set of J elements that are indexed by j. We first observe that since l is fixed, and all $j = 1, 2, \ldots, J$ are involved, the set of all $\mathbf{W}_o$ resulting from $\mathbf{W}_j\mathbf{W}_l$ is the same as the set of all $\mathbf{W}_j$ for $j = 1, 2, \ldots, J$. Also, the set of $\mathbf{w}_o$ is the same as the set of $\mathbf{w}_j$ for all j.

Recall that a composition of two elements of a space group is also an element of the space group: $\mathbf{W}_j\mathbf{w}_l + \mathbf{w}_j = \mathbf{w}_o$ when $\mathbf{W}_j\mathbf{W}_l = \mathbf{W}_o$. Hence, we obtain $(\mathbf{W}_j\mathbf{W}_l, \mathbf{W}_j\mathbf{w}_l - \mathbf{w}_j) = (\mathbf{W}_j\mathbf{W}_l, \mathbf{W}_j\mathbf{w}_l + \mathbf{w}_j - 2\mathbf{w}_j) = (\mathbf{W}_o, \mathbf{w}_o - 2\mathbf{w}_j)$. Combining these relations and Eq. (220), we obtain $\mathbf{w}_o - 2\mathbf{w}_j = -\mathbf{w}_o$, and hence $\mathbf{w}_o = \mathbf{w}_j$. Going back to the definition of $\mathbf{w}_o$, we observe $\mathbf{w}_o = \mathbf{W}_j\mathbf{w}_l + \mathbf{w}_j = \mathbf{w}_j$. Therefore, we eventually end up with

$$\mathbf{W}_j\mathbf{w}_l = \mathbf{0}, \text{ for all } \mathbf{G}_j \in \mathcal{G}_i, \tag{221}$$

where l is fixed. Recall that the $\mathbf{W}_j$'s are all isometries, meaning they preserve the length of vectors on which they operate. Therefore, the only possible choice for $\mathbf{w}_l$ that can satisfy Eq. (221) is $\mathbf{w}_l = \mathbf{0}$. The same arguments can be applied to prove that this condition $\mathbf{w}_l = \mathbf{0}$ is true for all $l = 1, 2, \ldots, J$. Hence, all the translation components of all the elements of $\mathcal{G}_i$ should be $\mathbf{0}$.

Next, conversely, we show that if all the translation components $\mathbf{w}_j$ of all $\mathbf{G}_j \in \mathcal{G}_i$ are zero, then all the estimates ρ_k produced by Algorithm 2 have the same space group $\mathcal{G}_i$ when the algorithm is initialized with a function with space group $\mathcal{G}_i$.

Assume that $\mathbf{w}_j = \mathbf{0}$ for all $\mathbf{G}_j \in \mathcal{G}_i$. Then, $\Lambda(\mathbf{h}^\mathbf{T}) = \Lambda(\mathbf{h}^\mathbf{T}\mathbf{W}_j)$ from Eq. (27). Using this condition, Q_k becomes

$$\begin{aligned}
Q_k(\mathbf{r}) &= \frac{1}{V}\sum_{\mathbf{h}} \Lambda(\mathbf{h}^\mathbf{T}) \exp(-2\pi i \mathbf{h}^\mathbf{T}\mathbf{r}) \\
&= \frac{1}{V}\sum_{\mathbf{h}\in\mathcal{I}(\mathcal{G}_t)} \sum_{j=1}^{J} \Lambda(\mathbf{h}^\mathbf{T}\mathbf{W}_j) \exp(-2\pi i \mathbf{h}^\mathbf{T}\mathbf{W}_j\mathbf{r}) \\
&= \frac{1}{V}\sum_{\mathbf{h}\in\mathcal{I}(\mathcal{G}_t)} \Lambda(\mathbf{h}^\mathbf{T}) \sum_{j=1}^{J} \exp(-2\pi i \mathbf{W}_j\mathbf{r}).
\end{aligned} \tag{222}$$

Hence $Q_k(\mathbf{r}) = Q_k(\mathbf{G}_l\mathbf{r}) = Q_k(\mathbf{W}_l\mathbf{r}),$ for all $\mathbf{G}_l \in \mathcal{G}_i$, since

$$Q_k(\mathbf{W}_l\mathbf{r}) = \frac{1}{V}\sum_{\mathbf{h}\in\mathcal{I}(\mathcal{G}_t)} \Lambda(\mathbf{h}^{\mathbf{T}})\sum_{j=1}^{J}\exp(-2\pi i\mathbf{W}_j\mathbf{W}_l\mathbf{r}), \tag{223}$$

and the group element composition $\mathbf{W}_j\mathbf{W}_l$ forms $\mathcal{G}_i$ back, just as in the Patterson space group analysis in the proof of Theorem 1. Thus, at every iteration, ρ_k and Q_k have the same space group for all k if all the translation components $\mathbf{w}_j$ of all $\mathbf{G}_j \in \mathcal{G}_i$ are zero, and the algorithm is initialized with an image with space group $\mathcal{G}_i$. This completes the proof of Theorem 2 in Section 2.

A.4 Proof of Corollary 2

We first prove Corollary 2(i). Let $R = \rho * P_\rho$. Let Ω denote the Fourier transform of R. By the convolution theorem of the Fourier transform [82], we obtain

$$\begin{aligned}
\Omega(\mathbf{h}^{\mathbf{T}}) &= \mathscr{F}\{R\} = \mathscr{F}\{\rho * P_\rho\} = |F(\mathbf{h}^{\mathbf{T}})|^2 F(\mathbf{h}^{\mathbf{T}}) \\
&= |F(\mathbf{h}^{\mathbf{T}}\mathbf{W}_j)|^2 F(\mathbf{h}^{\mathbf{T}}\mathbf{W}_j)\exp(2\pi i\mathbf{h}^{\mathbf{T}}\mathbf{w}_j), \text{ for all } \mathbf{G}_j \in \mathcal{G}_i \\
&= \Omega(\mathbf{h}^{\mathbf{T}}\mathbf{W}_j)\exp(2\pi i\mathbf{h}^{\mathbf{T}}\mathbf{w}_j), \text{ for all } \mathbf{G}_j \in \mathcal{G}_i,
\end{aligned} \tag{224}$$

where $\mathbf{G}_j = (\mathbf{W}_j, \mathbf{w}_j)$. The second to the last equality holds by Eq. (23). By taking the inverse Fourier transform of Ω, we can express R as

$$\begin{aligned}
R(\mathbf{r}) &= \frac{1}{V}\sum_{\mathbf{h}}\Omega(\mathbf{h}^{\mathbf{T}})\exp(-2\pi i\mathbf{h}^{\mathbf{T}}\mathbf{r}) \\
&= \frac{1}{V}\sum_{\mathbf{h}\in\mathcal{I}(\mathcal{G}_t)}\sum_{j=1}^{J}\Omega(\mathbf{h}^{\mathbf{T}}\mathbf{W}_j)\exp(-2\pi i\mathbf{h}^{\mathbf{T}}\mathbf{W}_j\mathbf{r}) \\
&= \frac{1}{V}\sum_{\mathbf{h}\in\mathcal{I}(\mathcal{G}_t)}\sum_{j=1}^{J}\Omega(\mathbf{h}^{\mathbf{T}})\exp(-2\pi i\mathbf{h}^{\mathbf{T}}\mathbf{w}_j)\exp(-2\pi i\mathbf{h}^{\mathbf{T}}\mathbf{W}_j\mathbf{r}), \\
&= \frac{1}{V}\sum_{\mathbf{h}\in\mathcal{I}(\mathcal{G}_t)}\Omega(\mathbf{h}^{\mathbf{T}})\sum_{j=1}^{J}\exp(-2\pi i(\mathbf{W}_j\mathbf{r}+\mathbf{w}_j)),
\end{aligned} \tag{225}$$

where $\mathcal{I}(\mathcal{G}_t)$ is defined the same as in the proof of Theorem 2. Now, we observe that

$$\begin{aligned}
R(\mathbf{W}_k\mathbf{r}+\mathbf{w}_k) &= \frac{1}{V}\sum_{\mathbf{h}\in\mathcal{I}(\mathcal{G}_t)}\Omega(\mathbf{h}^{\mathbf{T}})\sum_{j=1}^{J}\exp(-2\pi i(\mathbf{W}_j\mathbf{W}_k\mathbf{r}+\mathbf{W}_j\mathbf{w}_k+\mathbf{w}_j)), \\
&= R(\mathbf{r}), \text{ for all } \mathbf{G}_k \in \mathcal{G}_i.
\end{aligned} \tag{226}$$

The last equality is satisfied since the $(\mathbf{W}_j\mathbf{W}_k, \mathbf{W}_j\mathbf{w}_k + \mathbf{w}_j)$ is simply a group element composition and belongs to $\mathcal{G}_i$ for a fixed k. Moreover, for a fixed k, if the j's are distinct, the corresponding compositions are also distinct; if k is fixed and all j's are involved, all the compositions form the space group $\mathcal{G}_i$. This is true for all $k = 1, 2, \ldots, J$. Therefore, R has the same space group $\mathcal{G}_i$ as ρ.

Next, we prove Corollary 2(ii). Let the Fourier transforms of $\tilde{\rho} * P_\rho$ be Ψ. Note that this Ψ satisfies Eq. (27): $\Psi(\mathbf{h}^\mathbf{T}) = \Psi(\mathbf{h}^\mathbf{T}\mathbf{W}_j)\exp(-2\pi i\mathbf{h}^\mathbf{T}\mathbf{w}_j)$. Hence, arguments similar to those used in the proof of Theorem 2 prove this corollary.

APPENDIX B

SUPPLEMENTARY FOR CHAPTER IV

B.1 Proof of Theorem 1

Let f^* denote an estimate satisfying the Kuhn-Tucker conditions given in Eq. (8). For an arbitrary f that satisfies the nonnegativity constraint and is in the neighborhood of f^*, we want to show that

$$J(f^* + \Delta f) - J(f^*) > 0. \tag{227}$$

To do so, define $\Delta f = f - f^*$ and start with a Taylor expansion formula:

$$\begin{aligned} J(f^* + \Delta f) = J(f^*) \; &+ \; \sum_i \frac{\partial J(f^*)}{\partial f(x_i)} \Delta f(x_i) \\ &+ \; \frac{1}{2} \sum_i \sum_j \frac{\partial^2 D[S, R_{f^*}]}{\partial f(x_i) \partial f(x_j)} \Delta f(x_i) \Delta f(x_j) \; + \; o(\|\Delta f(x_i)\|^2), \end{aligned} \tag{228}$$

where o(·) is defined as in [80, p. 861]. We now move $D[S, R_{f^*}]$ to the left-hand side, and regroup the parameters on the right-hand side according to the sets $\mathcal{S}_1$ and $\mathcal{S}_2$. Then, we obtain

$$\begin{aligned} J(f^* + \Delta f) - J(f^*) = & \sum_{i \in \mathcal{S}_1} \frac{\partial J(f^*)}{\partial f(x_i)} \Delta f(x_i) \; + \; \sum_{i \in \mathcal{S}_2} \frac{\partial J(f^*)}{\partial f(x_i)} \Delta f(x_i) \\ & + \; \frac{1}{2} \sum_{i \in \mathcal{S}_1} \sum_{j \in \mathcal{S}_1} \frac{\partial^2 J(f^*)}{\partial f(x_i) \partial f(x_j)} \Delta f(x_i) \Delta f(x_j) \\ & + \; \frac{1}{2} \sum_{i \in \mathcal{S}_1} \sum_{j \in \mathcal{S}_2} \frac{\partial^2 J(f^*)}{\partial f(x_i) \partial f(x_j)} \Delta f(x_i) \Delta f(x_j) \\ & + \; \frac{1}{2} \sum_{i \in \mathcal{S}_2} \sum_{j \in \mathcal{S}_1} \frac{\partial^2 J(f^*)}{\partial f(x_i) \partial f(x_j)} \Delta f(x_i) \Delta f(x_j) \\ & + \; \frac{1}{2} \sum_{i \in \mathcal{S}_2} \sum_{j \in \mathcal{S}_2} \frac{\partial^2 J(f^*)}{\partial f(x_i) \partial f(x_j)} \Delta f(x_i) \Delta f(x_j) \\ & + \; o(\|\Delta f(x_i)\|^2). \end{aligned} \tag{229}$$

The index set $\mathcal{S}_2$ may be empty. However, we assume $\mathcal{S}_2$ is non-empty for generality. The proof can be easily modified for the case of empty $\mathcal{S}_2$, though. Now, by the definition of the index sets, the first term on the right-hand side is zero, and the second term is positive because all the parameters in the set $\mathcal{S}_2$ are on the boundary, and hence the $\Delta f(x_i)$ are always positive. We now inspect the fourth through the sixth terms on the right-hand side. Since we have only two $\Delta f(x_i)$ terms, we can always relate the two terms in a trivially linear way. In other words, for any $\Delta f(x_i)$ and $\Delta f(x_j)$ such that $i \in \mathcal{S}_1$ and $j \in \mathcal{S}_2$, there always exists a constant c_{ij} such that $\Delta f(x_i) = c_{ij}\Delta f(x_j)$. Using this relation, we conclude that

$$\begin{aligned}\frac{1}{2}\sum_{i\in\mathcal{S}_1}\sum_{j\in\mathcal{S}_2}\frac{\partial^2 J(f^*)}{\partial f(x_i)\partial f(x_j)}\Delta f(x_i)\Delta f(x_j) &= \frac{1}{2}\sum_{i\in\mathcal{S}_1}\sum_{j\in\mathcal{S}_2}\frac{\partial^2 J(f^*)}{\partial f(x_i)\partial f(x_j)}c_{ij}[\Delta f(x_j)]^2\\ &= o(\|\Delta f(x_i)\|). \end{aligned} \tag{230}$$

Therefore, the fourth term is dominated by the second term on the right-hand side of Eq. (229) for sufficiently small $||\Delta f(x_i)||$. Hence, we can neglect the fourth term. The same reasoning easily proves that the fifth and sixth terms are also negligible. As we have seen so far, all the terms on the right-hand side except for the third term are either positive or negligible. In the meantime, the second condition in Eq. (40) guarantees positivity of the third term. Therefore, we finally obtain the inequality in Eq. (227). This proves the theorem.

B.2 Derivation of The Second Partial Derivative

We start with the first derivative:

$$
\begin{aligned}
\frac{\partial^2 J(f)}{\partial f(x_i)\partial f(x_j)} &= \frac{\partial}{\partial f(x_j)}\left[2\sum_x f(x) - \sum_y \{f(x_i+y)+f(x_i-y)\}\frac{S(y)}{R_f(y)}\right] \\
&= 2 - \sum_y \frac{\partial}{\partial f(x_i)}\{f(x_i+y)+f(x_i-y)\}\frac{S(y)}{R_f(y)} \\
&= 2 - \sum_y S(y)\left\{\frac{f'(x_i+y)R_f(y) - f(x_i+y)R'_f(y)}{R_f^2(y)}\right. \\
&\quad \left. + \frac{f'(x_i-y)R_f(y) - f(x_i-y)R'_f(y)}{R_f^2(y)}\right\} \\
&= 2 - \sum_y S(y)\left\{\frac{\delta(x_i+y-x_j)}{R_f(y)} - \frac{f(x_i+y)\{f(x_j+y)+f(x_j-y)\}}{R_f^2(y)}\right. \\
&\quad \left. + \frac{\delta(x_i-y-x_j)}{R_f(y)} - \frac{f(x_i-y)\{f(x_j+y)+f(x_j-y)\}}{R_f^2(y)}\right\} \\
&= 2 + \sum_y \{f(x_i+y)f(x_j+y) + f(x_i+y)f(x_j-y)\}\frac{S(y)}{R_f(y)} \\
&\quad + \sum_y \{f(x_i-y)f(x_j+y) + f(x_i-y)f(x_j-y)\}\frac{S(y)}{R_f(y)} \\
&\quad - \left(\frac{S(x_j-x_i)}{R_f(x_j-x_i)} + \frac{S(x_i-x_j)}{R_f(x_i-x_j)}\right) \\
&= 2 + \sum_y \{f(x_i+y)+f(x_i-y)\}\{f(x_j+y)+f(x_j-y)\}\frac{S(y)}{R_f(y)} \\
&\quad - \left(\frac{2S(x_j-x_i)}{R_f(x_j-x_i)}\right).
\end{aligned} \tag{231}
$$

The last equality holds because an autocorrelation is symmetric.

APPENDIX C

SUPPLEMENTARY TABLES FOR CHAPTER VII

C.1 Convergence and Regularization Parameter Function Control Variables

Table 18: Iteration numbers at which the estimates converge, for unconstrained reconstructions from noiseless measurements.

Pattern	Convergence
Gaussian-like	10000
Rectangle	1000000
Triangle	100000000
Double Gaussian-like	100000000
Double triangle	100000000

Table 19: Iteration numbers at which the estimates converge and the parameter function control variables c_{max} and c_{min}, for regularized reconstructions from noiseless measurements when Good's roughness penalty is applied.

Pattern	Convergence	c_{min}	c_{max}
Gaussian-like	3000	1×10^{-30}	2×10^{-13}
Rectangle	5000	1×10^{-50}	2×10^{-11}
Triangle	4000	1×10^{-50}	3×10^{-12}
Double Gaussian-like	1200000	2×10^{-16}	2×10^{-15}
Double triangle	1000000	2×10^{-16}	2×10^{-15}

Table 20: Iteration numbers at which the estimates converge and the parameter function control variables c_{max} and c_{min}, for regularized reconstructions from noiseless measurements when our entropy-like penalty is applied.

Pattern	Convergence	c_{min}	c_{max}
Gaussian-like	2000	2×10^{-16}	2×10^{-15}
Rectangle	5000	1×10^{-60}	3×10^{-12}
Triangle	25000	1×10^{-60}	1×10^{-13}
Double Gaussian-like	2000000	7×10^{-17}	9×10^{-16}
Double triangle	2000000	7×10^{-17}	9×10^{-16}

Table 21: Iteration numbers at which the estimates converge, for unconstrained reconstructions from noisy measurements.

Pattern	Convergence (low noise)	Convergence (high noise)
Gaussian-like	1000000	240000
Rectangle	1700000	800000
Triangle	2100000	1200000
Double Gaussian-like	2400000	1700000
Double triangle	3500000	1100000

Table 22: Iteration numbers at which the estimates converge and the parameter function control variables c_{max} and c_{min}, for regularized reconstructions from noisy measurements when Good's roughness penalty is applied, and the noise level is low: $k_n = 10^{-13}$.

Pattern	Convergence	c_{min}	c_{max}
Gaussian-like	5000	1×10^{-60}	5×10^{-11}
Rectangle	4000	1×10^{-15}	1×10^{-10}
Triangle	3000	1×10^{-14}	3×10^{-11}
Double Gaussian-like	100000	1×10^{-60}	5×10^{-13}
Double triangle	100000	1×10^{-60}	4×10^{-13}

Table 23: Iteration numbers at which the estimates converge and the parameter function control variables c_{max} and c_{min}, for regularized reconstructions from noisy measurements when Good's roughness penalty is applied, and the noise level is high: $k_n = 10^{-12}$.

Pattern	Convergence	c_{min}	c_{max}
Gaussian-like	2000	1×10^{-60}	2.5×10^{-10}
Rectangle	2000	1×10^{-14}	1×10^{-10}
Triangle	1000	1×10^{-14}	1.5×10^{-10}
Double Gaussian-like	50000	1×10^{-60}	8×10^{-12}
Double triangle	30000	1×10^{-60}	6×10^{-12}

Table 24: Iteration numbers at which the estimates converge and the parameter function control variables c_{max} and c_{min}, for regularized reconstructions from noisy measurements when our entropy-like is applied, and the noise level is low: $k_n = 10^{-13}$.

Pattern	Convergence	c_{min}	c_{max}
Gaussian-like	10000	1×10^{-60}	3×10^{-12}
Rectangle	6000	1×10^{-15}	5×10^{-12}
Triangle	6000	1×10^{-15}	3×10^{-12}
Double Gaussian-like	140000	1×10^{-40}	2×10^{-13}
Double triangle	200000	1×10^{-30}	2×10^{-13}

Table 25: Iteration numbers at which the estimates converge and the parameter function control variables c_{max} and c_{min}, for regularized reconstructions from noisy measurements when our entropy-like is applied, and the noise level is high: $k_n = 10^{-12}$.

Pattern	Convergence	c_{min}	c_{max}
Gaussian-like	2000	1×10^{-60}	1.4×10^{-11}
Rectangle	2000	1×10^{-14}	9×10^{-12}
Triangle	1000	1×10^{-14}	1.2×10^{-11}
Double Gaussian-like	50000	1×10^{-40}	1.5×10^{-12}
Double triangle	40000	1×10^{-60}	1.4×10^{-12}

Table 26: Iteration numbers at which the estimates converge and the parameter function control variables c_{max} and c_{min}, for over-regularized reconstructions from noisy measurements when Good's roughness penalty is applied, and the noise level is low: $k_n = 10^{-13}$.

Pattern	Convergence	c_{min}	c_{max}
Gaussian-like	1000	1×10^{-60}	2.5×10^{-10}
Rectangle	2000	1×10^{-14}	1×10^{-10}
Triangle	1000	1×10^{-14}	1.5×10^{-10}
Double Gaussian-like	20000	1×10^{-60}	8×10^{-12}
Double triangle	20000	1×10^{-60}	6×10^{-12}

Table 27: Iteration numbers at which the estimates converge and the parameter function control variables c_{max} and c_{min}, for under-regularized reconstructions from noisy measurements when Good's roughness penalty is applied, and the noise level is high: $k_n = 10^{-12}$.

Pattern	Convergence	c_{min}	c_{max}
Gaussian-like	3000	1×10^{-60}	5×10^{-11}
Rectangle	2000	1×10^{-15}	1×10^{-10}
Triangle	2000	1×10^{-14}	3×10^{-11}
Double Gaussian-like	50000	1×10^{-60}	5×10^{-13}
Double triangle	50000	1×10^{-60}	4×10^{-13}

Table 28: Iteration numbers at which the estimates converge and the parameter function control variables c_{max} and c_{min}, for over-regularized reconstructions from noisy measurements when our entropy-like is applied, and the noise level is low: $k_n = 10^{-13}$.

Pattern	Convergence	c_{min}	c_{max}
Gaussian-like	2000	1×10^{-60}	1.4×10^{-11}
Rectangle	2000	1×10^{-14}	9×10^{-12}
Triangle	1000	1×10^{-14}	1.2×10^{-11}
Double Gaussian-like	20000	1×10^{-40}	1.5×10^{-12}
Double triangle	20000	1×10^{-60}	1.4×10^{-12}

Table 29: Iteration numbers at which the estimates converge and the parameter function control variables c_{max} and c_{min}, for under-regularized reconstructions from noisy measurements when our entropy-like penalty is applied, and the noise level is high: $k_n = 10^{-12}$.

Pattern	Convergence	c_{min}	c_{max}
Gaussian-like	7000	1×10^{-60}	3×10^{-12}
Rectangle	5000	1×10^{-15}	5×10^{-12}
Triangle	7000	1×10^{-15}	3×10^{-12}
Double Gaussian-like	120000	1×10^{-40}	2×10^{-13}
Double triangle	120000	1×10^{-30}	2×10^{-13}

REFERENCES

[1] A, A., ed., *Regularization, Uniqueness and Existence of Solutions of Volterra Equations of the First Kind.* The Netherlands: VSP, 1998.

[2] A, N. J. and F, S. D., "Frequency domain methods for volterra equations," *Adv. Math.*, vol. 22, pp. 278–304, 1976.

[3] Amato, U. and Hughes, W., "Maximum entropy regularization of fredholm integral equations of the first kind," *Inverse Problems*, vol. 7, pp. 793–808, 1991.

[4] Bartle, R. G., ed., *The Elements of Real Analysis.* Wiley, 1976.

[5] Besag, J., "Spatial interaction and the statistical analysis of lattice systems (with discussion)," *J. R. Statist. Soc. Ser. B*, vol. 36, pp. 192–236, 1974.

[6] Besag, J., "On the statistical analysis of dirty pictures (with discussion)," *J. R. Statist. Soc. Ser. B*, vol. 48, pp. 259–302, 1986.

[7] Bojarski, N. N., "Inverse black body radiation," *IEEE Trans. Antennas Propag.*, vol. AP-30, no. 4, pp. 778–780, 1982.

[8] Bojarski, N. N., "Closed form approximations to the inverse black body radiation problem," *IEEE Trans. Antennas Propag.*, vol. AP-32, no. 4, pp. 415–418, 1984.

[9] Borchardt-Ott, W., ed., *Crystallography.* Springer-Verlag, 1995.

[10] Byrne, C. L., "Iterative image reconstruction algorithms based on cross-entropy minimization," *IEEE Trans. Image Process.*, vol. 2, pp. 96–103, 1993.

[11] Byrne, C. L., "Iterative algorithms for deblurring and deconvolution with constraints," *Inverse Problems*, vol. 14, pp. 1455–1467, 1998.

[12] Byrne, C. L., "A unified treatment of some iterative algorithms in signal processing and image reconstruction," *Inverse Problems*, vol. 20, pp. 103–120, 2004.

[13] Chan, R. H., Chan, T. F., and Wong, C. K., "Cosine transform based preconditioners for total variation deblurring," *IEEE Trans. Image Process.*, vol. 8, pp. 1472–1478, 1999.

[14] Chan, T. F., Golub, G. H., and Mulet, P., "A nonlinear primal-dual method for total variation-based image restoration," *SIAM J. Sci. Comput.*, vol. 20, pp. 1964–1977, 1999.

[15] Chan, T. F. and Wong, C., "Total variation blind deconvolution," *IEEE Trans. Image Process.*, vol. 7, no. 3, pp. 370–375, 1998.

[16] Chan, T. F. and Wong, C. K., "Convergence of the alternating minimization algorithm for blind deconvolution," *Linear Algebra and its Applications*, vol. 316, pp. 259–286, 2000.

[17] Chan, T. F. and Wong, C. K., "Multichannel image deconvolution by total variation regularization," in *Proc. to the SPIE Symposium on Advanced Signal Processing: Algorithms, Architectures, and Implementations (Ed.:F. Luk.)*, vol. 3162, July 1997.

[18] Chen, N. and Li, G., "Theoretical investigation on the inverse black body radiation problem," *IEEE Trans. Antennas Propag.*, vol. 38, no. 8, pp. 1287–1290, 1990.

[19] Combettes, P. L. and Luo, J., "An adaptive level set method for nondifferntiable constrained image recovery," *IEEE Trans. Image Process.*, vol. 11, pp. 1295–1304, 2002.

[20] Combettes, P. L. and Pesquet, J. C., "Image restoration subject to a total variation constraint," *IEEE Trans. Image Process.*, vol. 13, pp. 1213–1222, 2004.

[21] Cover, T. M., "An algorithm for maximizing expected log investment return," *IEEE Trans. Inform. Theory*, vol. 30, pp. 369–373, 1984.

[22] Cover, T. M. and Thomas, J. A., eds., *Elements of Information Theory.* Wiley, 1991.

[23] Csiszár, I., "Why least squares and maximum entropy? — an axiomatic approach to inverse problems," *Ann. Stat.*, vol. 19, pp. 2033–2066, 1991.

[24] Dempster, A., Laird, N., and Rubin, D., "Maximum likelihood from incomplete data via the em algorithm," *J. R. Statist. Soc. B*, vol. 39, pp. 1–37, 1977.

[25] Dodge, Y., *Statistical Data Analysis Based on the L_1 Norm and Related Methods.* North-Holland, Amsterdam, 1987.

[26] Donoho, D. L., Johnstone, I. M., Hoch, J. C., and Stern, A. S., "Maximum entropy and the nearly black object," *J. R. Statist. Soc. B*, vol. 54, pp. 41–81, 1992.

[27] Dose, V., Fauster, T., and Scheidt, H., "Isochromat and sxaps studies of empty electronic states in chromium, iron and nickel," *J. Phys. F.: Metal Phys.*, vol. 11, pp. 1801–1809, 1981.

[28] Dou, L. and Hodgeon, R. J. W., "Maximum entropy method in inverse black body radiation problem," *J. Appl. Phys.*, vol. 71, pp. 3159–3163, 1992.

[29] Dove, E. L., "Signal-to-noise ratio." `http://www.engineering.uiowa.edu/∼bme_285/Lecture`

[30] Egert, E. and Sheldrick, G. M., "Search for a fragment of known geometry by integrated patterson and direct methods," *Acta Cryst.*, vol. A41, pp. 262–268, 1985.

[31] Fessler, J. and Hero, A., "Space-alternating generalized expectation-maximization algorithm," *IEEE Trans. Signal Process.*, vol. 42, pp. 2664–2677, 1994.

[32] Fienup, J. R., "Reconstruction of an object from the modulus of its fourier transform," *Opt. Lett.*, vol. 3, pp. 27–29, 1978.

[33] Fienup, J. R., "Phase retrieval algorithms: a comparison," *Appl. Opt.*, vol. 21, pp. 2758–2769, 1982.

[34] Fienup, J. R. and Wackerman, C. C., "Phase-retrieval stagnation problems and solutions," *J. Opt. Soc. Am. A*, vol. 3, pp. 1897–1907, 1986.

[35] Fleischer, G., Gorenflo, R., and Hofmann, B., "On the autoconvolution equation and total variation constraints," *Z. angew. Math. Mech.*, vol. 79, pp. 149–159, 1999.

[36] Fleischer, G. and Hofmann, B., "On inversion rates for the autoconvolution equation," *Inverse Problems*, vol. 12, pp. 419–435, 1996.

[37] Fozunbal, M., McLaughlin, S. W., and Schafer, R. W., "A bijection property of flat-fading channels and its application to the capacity problem." submitted to *IEEE Trans. Inform. Theory*, 2003.

[38] Good, I. J., "Maximum entropy for hypothesis formulation, especially for multidimensional contingency tables," *Ann. Math. Stat.*, vol. 34, pp. 911–934, 1963.

[39] Good, I. J. and Gaskins, R. A., "Nonparametric roughness penalties for probability densities," *Biometrica*, vol. 58, pp. 255–277, 1971.

[40] Gorenflo, R. and Hofmann, B., "On autoconvolution and regularization," *Inverse Problems*, vol. 10, pp. 353–373, 1994.

[41] Green, P. J., "On use of the em for penalized likelihood estimation," *J. R. Statist. Soc. B*, vol. 52, no. 3, pp. 443–452, 1990.

[42] Green, P., "Bayesian reconstruction from emission tomography data using a modified em algorithm," *IEEE Trans. Med. Im.*, vol. 9, pp. 84–93, 1990.

[43] Hamermesh, M., ed., *Group Theory and Its Application to Physical Problems.* Dover, New York, 1962.

[44] Hamid, M., "Inverse black body radiation at microwave frequencies," *IEEE Trans. Antennas Propag.*, vol. AP-31, no. 5, pp. 810–812, 1983.

[45] Harker, D., "The application of the three dimensional patterson method and the crystal structures of proustite ag_3ass_3 and pyrargyrite ag_3sbs_3.," *J. Chem. Phys.*, vol. 4, pp. 381–390, 1936.

[46] Hermann, U. and Noll, D., "Adaptive image reconstruction using information measures," *SIAM J. Control Optim.*, vol. 38, pp. 1223–1240, 2000.

[47] Hobson, A. and Cheng, B., "A comparison of the shannon and kullback information measures," *J. Stat. Phys.*, vol. 7, pp. 301–310, 1973.

[48] Hunter, J. D., "An improved closed-form approximation to the inverse black body radiation problem at microwave frequencies," *IEEE Trans. Antennas Propag.*, vol. AP-34, no. 2, pp. 261–262, 1986.

[49] Jamshidian, M. and Jennrich, R. I., "Conjugate gradient acceleration of the em algorithm," *J. Am. Statist. Soc.*, vol. 88, pp. 221–228, 1993.

[50] Jamshidian, M. and Jennrich, R. I., "Acceleration of the em algorithm by using quasi-newton methods," *J. R. Statist. Soc. B*, vol. 59, pp. 569–587, 1997.

[51] Janno, J., "Lavrent'ev regularization of ill-posed problems containing nonlinear near-to-monotone operators with application to autoconvolution equation," *Inverse Problems*, vol. 16, pp. 333–348, 2000.

[52] Johnson, R. W., "Axiomatic characterization of the directed divergences and their linear combinations," *IEEE Trans. Inform. Theory*, vol. 25, pp. 129–132, 1979.

[53] Johnson, R. W., "Comments on 'prior probability and uncertainty'," *IEEE Trans. Inform. Theory*, vol. 25, pp. 129–132, 1979.

[54] Jones, L. K. and Byrne, C. L., "General entropy criteria for inverse problems, with applications to data compression, pattern classification, and cluster analysis," *IEEE Trans. Inform. Theory*, vol. 36, pp. 23–30, 1990.

[55] Jonsson, E., Huang, S., and Chan, T., "Total variation regularization in positron emission tomography." U.C.L.A. computational and applied mathematics reports, 98-48, 1998.

[56] Joshi, S. and Miller, M. I., "Maximum *a posteriori* estimation with good's roughness for three-dimensional optical-sectioning microscopy," *J. Opt. Soc. Am. A*, vol. 10, pp. 1078–1085, 1993.

[57] K, L. P., "Future-sequential regularization methods for ill-posed volterra equations," *J. Math. Anal. Appl.*, vol. 195, pp. 469–494, 1995.

[58] Kashyap, R. L., "Prior probability and uncertainty," *IEEE Trans. Inform. Theory*, vol. 17, pp. 641–650, 1971.

[59] Keeling, S. L., "Total variation based convex filters for medical imaging," *Appl. Math. Comput.*, vol. 139, pp. 101–119, 2003.

[60] Keith, M. and Gregory, A., *Black Body Radiation*. Egglescliffe School, 2005.

[61] Kim, Y. and Jaggard, D. L., "Inverse black body radiation: An exact closed-form solution," *IEEE Trans. Antennas Propag.*, vol. AP-33, no. 7, pp. 797–800, 1985.

[62] Kullback, S., ed., *Information Theory and Statistics*. Wiley, 1959.

[63] Kullback, S., "A lower bound for discrimination information in terms of variation," *IEEE Trans. Inform. Theory*, vol. 13, pp. 126–127, 1967.

[64] Kullback, S. and Leibler, R. A., "On information and sufficiency," *Ann. Math. Statist.*, vol. 22, no. 1, pp. 79–86, 1971.

[65] Landl, G. and Anderssen, R. S., "Non-negative differentially constrained entropy-like regularization," *Inverse Problems*, vol. 12, pp. 35–53, 1996.

[66] Lantéri, H., Roche, M., and Aime, C., "Penalized maximum likelihood image restoration with positivity constraints: Multiplicative algorithms," *Inverse Problems*, vol. 18, pp. 1397–1419, 2002.

[67] Lanterman, A. D., "A new way to regularize maximum likelihood estimates for emission tomography with good's roughness penalty." ESSRL Monograph, presented at the IEEE Region 5 conference, San Antonio, Texas, April 1992.

[68] LANTERMAN, A. D., "Statistical imaging in radio astronomy via an expectation-maximization algorithm for structured covariance estimation." in Statistical Methods in Imaging: In Medicine, Optics, and Communication, a festschrift in honor of Donald L. Snyder's 65th birthday, Ed. J.A. O'Sullivan, Springer-Verlag, to appear.

[69] LI, Y. and SANTOSA, F., "A computational algorithm for minimizing total variation in image restoration," *IEEE Trans. Image Process.*, vol. 5, pp. 987–995, 1996.

[70] LIU, C. and RUBIN, D. B., "The ecme algorithm: A simple extension of em and ecm with faster monotone convergence," *Biometrika*, vol. 81, pp. 633–648, 1994.

[71] LIU, C., RUBIN, D. B., and WU, Y. N., "Parameter expansion to accelerate em: The px-em algorithm," *Biometrika*, vol. 85, pp. 755–770, 1998.

[72] LUCY, L., "An iterative technique for the rectification of observed distributions," *The Astronomical Journal*, vol. 79, pp. 745–754, 1974.

[73] LUENBERGER, D. G., ed., *Optimization by Vector Space Methods*. John Wiley, New York, 1968.

[74] MACKINNON, R. F., "Minimum cross-entropy noise reduction in images," *J. Opt. Soc. Am. A*, vol. 6, pp. 739–747, 1989.

[75] MCCARTHY, A. W. and MILLER, M. I., "Maximum likelihood spect in clinical computation times using mesh-connected parallel computers," *IEEE Trans. Med. Imaging*, vol. 10, no. 3, pp. 426–436, 1991.

[76] MEAD, L. R., "Approximate solution of fredholm integral equations by the maximum-entropy method," *J. Appl. Phys.*, vol. 27, pp. 2903–2907, 1986.

[77] MILLANE, R. P., "Phase problems for periodic images: Effects of support and symmetry," *J. Opt. Soc. Am. A*, vol. 10, pp. 1037–1045, 1993.

[78] MILLER, M. I. and ROYSAM, B., "Bayesian image reconstruction for emission tomography incorporating good's roughness prior on massively parallel processors," in *Proc. Natl. Acad. Sci. U.S.A.*, vol. 88, pp. 3223–3227, 1991.

[79] MILLER, M. I. and SNYDER, D. L., "The role of likelihood and entropy in incomplete-data problems: Applications to estimating point-process intensities and toeplitz constrained covariances," *Proc. IEEE*, vol. 75, no. 7, pp. 892–907, 1987.

[80] MOON, T. K. and STIRLING, W. C., eds., *Mathematical Methods and Algorithms for signal processing*. Prentice Hall, 1999.

[81] MORRIS, G. and HARKNESS, L., eds., *Airborne Pulsed Doppler Radar*. Artech House, 1996.

[82] OPPENHEIM, A. V. and SCHAFER, R. W., eds., *Discrete Time Signal Processing*. Prentice Hall, 1999.

[83] O'SULLIVAN, J. A., "Roughness penalities on finite domains," *IEEE Trans. Image Process.*, vol. 4, no. 9, pp. 1258–1268, 1995.

[84] O'Sullivan, J. A. and Snyder, D. L., "Deterministic em algorithms with penalties." submitted to International Symposium on Information Theory, Whistler, British Columbia, September 1995.

[85] Patterson, A. L., "A fourier series method for the determination of the components of interatomic distances in crystals," *Phys. Rev.*, vol. 46, pp. 372–376, 1934.

[86] Patterson, A. L., "A direct method for the determination of the components of interatomic distances in crystals," *Z. Krist.*, vol. A90, pp. 517–542, 1935.

[87] R, P., "The galerkin scheme for lavrentiev m-times iterated method to solve linear accretive volterra integral equations of the first kind," *BIT*, vol. 37, pp. 404–423, 1997.

[88] Richardson, W. H., "Bayesian-based iterative method of image restoration," *J. Opt. Soc. Am. A*, vol. 62, pp. 55–59, 1972.

[89] Royden, H. L., ed., *Real Analysis.* Prentice Hall, 1988.

[90] Rudin, L. I., Osher, S., and Fatemi, E., "Nonlinear total variation based noise removal algorithms," *Physica D*, vol. 60, pp. 259–268, 1992.

[91] Schenk, H., ed., *Direct Methods of Solving Crystal Structures.* Plenum Press, New York, 1991.

[92] Schulz, T. J. and Snyder, D. L., "Imaging a randomly moving object from quantum-limited data: Applications to image recovery from second- and third-order autocorrelations," *J. Opt. Soc. Am. A*, vol. 8, pp. 801–807, 1991.

[93] Schulz, T. J. and Snyder, D. L., "Image recovery from correlations," *J. Opt. Soc. Am. A*, vol. 9, pp. 1266–1272, 1992.

[94] Shepp, L. A. and Vardi, Y., "Maximum likelihood reconstruction for emission tomography," *IEEE Trans. Med. Im.*, vol. MI-1, pp. 113–122, 1982.

[95] Sherman, M. B., Brink, J., and Chiu, W., "Performance of a slow-scan ccd camera for macromolecular imaging in a 400 kv electron cryomicroscope," *Inverse Problems*, vol. 27, pp. 129–139, 1996.

[96] Shmueli, U., "Theories and techniques of crystal structure determination." draft with the tentative title, School of Chemistry, Tel Aviv University. Part of this draft is available at http://crystal.tau.ac.il/xtal/chapter6/node9.html.

[97] Shore, J. E., "Minimum cross-entropy spectral analysis," Tech. Rep. NRL Memo. Rep. 3921, Naval Res. Lab., Washington, DC 20375, 1979.

[98] Shore, J. E. and Johnson, R. W., "Axiomatic derivation of the principle of maximum entropy and the principle of maximum cross-entropys," *IEEE Trans. Inform. Theory*, vol. 26, pp. 26–37, 1980.

[99] Silverman, B. W., Jones, M. C., Wilson, J. D., and Nychka, D. W., "A smoothed em approach to indirect estimation problems, with particular, references to stereology and emission tomography," *J. R. Statist. Soc. B*, vol. 52, no. 2, pp. 271–324, 1990.

[100] Snyder, D. L., Hammoud, A. M., and White, R. L., "Image recovery from data acquired with a charge-coupled-device camera," *J. Opt. Soc. Am. A*, vol. 10, pp. 1014–1023, 1993.

[101] Snyder, D. L., Helstrom, C. W., Lanterman, A. D., and White, R. L., "Compensation for read-out noise in charge-coupled-device images," *J. Opt. Soc. Am. A*, vol. 12, pp. 272–283, 1995.

[102] Snyder, D. L., Lanterman, A. D., and Miller, M. I., "An extension of good's roughness penalty for nonparametric density-estimation." Proc. 30th Annual Allerton Conf. on Communication, Control, and Computing, Univ. of Illinois, Urbana IL, Sept. 1993.

[103] Snyder, D. L., Lanterman, A. D., and Miller, M. I., "A regularizing images in emission tomography via an extention of good's roughness penalty." ESSRL Monograph, presented at the IEEE Medical Imaging Conference, Orlando, Florida, Nov. 1993.

[104] Snyder, D. L. and Miller, M. I., eds., *Random Point Processes in Time and Space*. Springer-Verlag, 1991.

[105] Snyder, D. L., Miller, M. I., Thomas, L. J., and Politte, D. G., "Noise and edge artifacts in maximum-likelihood reconstructions for emission tomography," *IEEE Trans. Med. Imaging*, vol. 6, no. 3, pp. 228–238, 1987.

[106] Snyder, D. L., O'Sullivan, J. A., Whiting, B. R., Murphy, R. J., Benac, J., Cataldo, J. A., Politte, D. G., and Williamson, J. F., "Deblurring subject to nonnegativity constraints when known functions are present with application to object-constrained computerized tomogrpahy," *IEEE Trans. Med. Imaging*, vol. 20, pp. 1009–1017, 2001.

[107] Snyder, D. L., Schulz, T., and O'Sullivan, J., "Deblurring subject to nonnegativity constraint," *IEEE Trans. Signal Process.*, vol. 40, pp. 1143–1150, 1992.

[108] Stout, G. H. and Jenson, L. H., eds., *X-Ray Structure Determination*. Macmillan, 1968.

[109] Sun, X. and Jaggard, D. L., "The inverse blackbody radiation problem: A regularization solution," *J. Appl. Phys.*, vol. 62, pp. 4382–4386, 1987.

[110] T. Hahn, E., ed., *International Tables for Crystallography, Brief Teaching Edition of Volume A: Space-group Symmetry*. Kluwer Academic Publishers, London, 2002.

[111] Talyor, C. A., "Physics education," *Institute of Physics Publishing*, vol. 2, pp. 276–277, 1967.

[112] Tan, X., Yang, G., Gu, B., and Dong, B., "Numerical investigation of the inverse blackbody radiation problem," *J. Opt. Soc. Am. A*, vol. 11, pp. 1068–1072, 1994.

[113] Teboul, S., Blanc-Féraud, L., Aubert, G., and Barlaud, M., "Variational approach for edge-preserving regularization using coupled pde's," *IEEE Trans. Image Process.*, vol. 7, pp. 387–397, 1998.

[114] VARDI, Y. and LEE, D., "From image deblurring to optimal investments: Maximum likelihood solutions for positive linear inverse problems," *J. R. Statist. Soc. B*, vol. 55, pp. 569–612, 1993.

[115] WONDRATSCHEK, H., "Matrices, mappings, and crystallographic symmetry," 2002.

[116] WOOLFSON, M. M., ed., *An Introduction to X-ray Crystallography.* Cambridge University Press, U.K., 1997.

[117] XIANXI, D. and JIQIONG, D., "On unique existence theorem and exact solution formula of the inverse black-body radiation problem," *IEEE Trans. Antennas Propag.*, vol. 40, no. 3, pp. 237–260, 1992.

VITA

Kerkil Choi received his B.S. Degree in Electronic Engineering from Inha University, Incheon, Korea in 2000 and his M.S. Degree in Electrical and Computer Engineering from the University of Florida at Gainsville in 2002. His master's thesis focused on emission tomography. He plans to receive his Ph.D. Degree in Electrical and Computer Engineering from Georgia Institute of Technology in 2005. His main theoretical interests lie in the realm of statistical signal and image processing, with an emphasis on iterative reconstruction algorithms and regularization techniques. His applied interests include phase retrieval (particularly in x-ray crystallography) and linear inverse problems.